VISIT: Virtual-Targeted Sequential Training with Hierarchical Federated Learning on Non-IID Data

Kung-Hao Chang, Te-Chuan Chiu, and Jang-Ping Sheu
Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
E-mail: s110062519@m110.nthu.edu.tw, theochiu@cs.nthu.edu.tw and sheujp@cs.nthu.edu.tw

***Abstract*—Recently, Federated Learning (FL) has realized Artificial Intelligence of Things (AIoT) applications to train a shared model while preserving user privacy collectively. However, the legacy FL framework performance is fundamentally threatened by scale limitation, non-independent and identically distributed (non-IID) data, and communication costs. Therefore, we propose VIrtual-targeted SequentIal Training with Hierarchical Federated Learning (VISIT), a novel framework to systematically distribute clients to suitable clusters for balancing data distributions among all FL subgroups. To the best of our knowledge, this work is the first attempt to introduce a *Virtual Target* concept along with a key metric, *Virtual Target Similarity* (VTS), to quantify the data harmonization in the whole HFL system. Based on our insightful *Client Set* arranging strategy, VISIT can wisely select each FL subgroup member to optimize diversity within each *Client Set* and similarity across different clusters while preserving user privacy. Numerical results demonstrate that VISIT improves accuracy by 41% and reduces total communication rounds by 82% compared to other state-of-the-art baselines with non-IID data on EMNIST and CIFAR-10 datasets.**

***Index Terms*—Hierarchical Federated Learning, Sequential Training, Clustering, non-IID.**

I. Introduction

Recently, Deep Learning (DL) has gained popularity in boosting various AIoT applications by training on high-performance computing platforms in the cloud with centralized repositories of vast datasets [1]. However, this remote training procedure involves sending the raw data generated by end devices to the cloud server, inevitably raising privacy concerns. To tackle this issue, Federated Learning (FL), a decentralized learning framework, has emerged as a privacy-preserving solution for the existing centralized training [2]. Nevertheless, FL naturally needs to face the following challenges at the cost of obtaining privacy: 1) *Performance bottleneck*: a legacy single-server layout fundamentally restricts the global model's performance due to training scale limitation. 2) *non-independent and identically distributed (non-IID) issue*: raw data kept at each client side to ensure user's privacy, potentially leading to significant model accuracy degradation and extending the convergence time. 3) *Communication overhead*: an intuitive strategy to mitigate non-IID effects, mostly triggering additional communication costs between the edge server and clients for improving model convergence.

To tackle the above problems, Zhong *et al.* [3] introduced a parallelizable FL algorithm that leverages multiple parameter servers (PSes) to boost convergence rates and communication efficiency within resource-constrained FL scenarios. Besides,

This work was supported by the National Science and Technology Council, Taiwan, under grant 112-2221-E-007-046-MY3 and 112-2222-E-007-002-MY3, and by Qualcomm Technologies, Inc. under grant SOW NAT-487844.

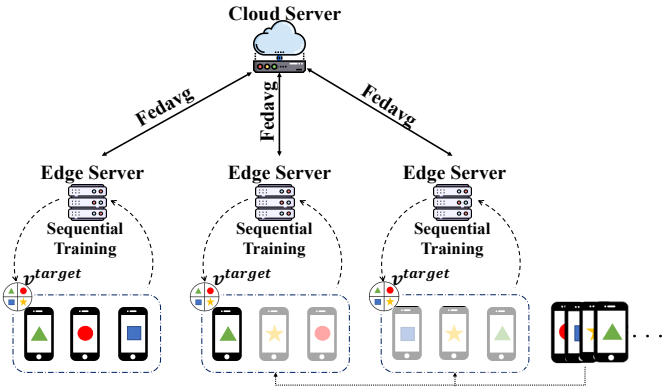


Fig. 1: VISIT architecture. The color patterns on the clients indicate the data distribution over each client.

Liu *et al.* [4] introduced a client-edge-cloud Hierarchical Federated Learning (HFL) framework enabling multiple edge servers to engage in partial model aggregation. The HFL system accommodates many clients owing to its hierarchical design, handling a tradeoff between communication and computation. Additionally, Mhaisen *et al.* [5] further improved HFL performance by employing synchronization algorithms customized to the communication characteristics of different layers. Based on the multi-layer design, HFL with a solitary central server fundamentally circumvents performance bottlenecks and reduces communication costs compared with legacy FL architecture. However, the side effect of multi-layer design exacerbates the non-IID issue instead. After the model aggregation at the bottom layer of each sub-FL group, those diverse models will amplify the non-IID chain effect to the topper-layer cloud server and make the global model merging hard to converge. Therefore, HFL must carefully address the non-IID issue with multi-layer structures.

Several FL research [6]–[13] have proposed clustering and different aggregation mechanisms to consider non-IID data in model training. Sattler *et al.* [10] proposed the concept of Clustered Federated Learning (CFL) to iteratively divide clients into small clusters based on the similarity observed in their model updates to improve the global model merging. Briggs *et al.* [11] designed a hierarchical clustering algorithm based on clients' cosine distance from the global model. Such a clustering approach improved the training performance of each cluster with its local aggregated model; however, they lacked awareness of the global model. Besides, other researchers [12], [13] studied clustering techniques to group clients with disparate data distributions for accumulating uniform distribution within each cluster with a better global model performance. Wang *et al.* [12] proposed a method that intelligently selects client

ICC 2024 - IEEE International Conference on Communications | 978-1-7281-9054-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICC51166.2024.10622579

devices in each FL round to counteract the bias introduced by non-IID data and accelerate convergence. Zaccone *et al.* [13] introduced FedSeq, an algorithm crafted to tackle statistical heterogeneity by employing sequential training among all FL subgroups. Clients with distinct data patterns are grouped to simulate the appearance of a larger and more uniform dataset while preserving privacy. However, the above strategy fails to achieve a balanced and diversified distribution in all FL subgroups, potentially leading to model bias and eventually dropping global model performance.

In this paper, we advocate HFL framework and adopt sequential training technique to optimize global model performance, especially under non-IID data distributions. We need to address the following key issues: 1) Estimating each client's distribution to form the cluster while preserving privacy. 2) Organizing *Client Set* to mitigate performance degradation influenced by non-IID data 3) Balancing the tradeoff between the global model performance and communication cost in the HFL system. Therefore, we propose **VI**rtual-targeted **S**equent**I**al **T**raining with HFL (VISIT), a novel framework to systematically distribute clients to suitable clusters for balancing data distributions among all FL subgroups. Our design strategy is to minimize the Intra *Client Set* Similarity (Intra-CS) among all clients in the same cluster (i.e., increase intra-cluster diversity) while enlarging Inter *Client Set* Similarity (Inter-CS) between different FL subgroups (i.e., preserve more IID subgroups). We adopt sequential training within each *Client Set* to mitigate the negative impact of non-IID on FL. To alleviate the potential model bias towards certain outliers, we introduce *Virtual Target*, a predefined uniform target, to assist *Client Set* forming while realizing the above design strategy. Besides, apart from Intra-CS and Inter-CS, we propose a key metric, *Virtual Target Similarity (VTS)*, as the mean similarity between *Client Set* and *Virtual Target* to quantify the data harmonization in the whole HFL system. The optimized VTS policy wisely chooses each FL subgroup member to balance diversity within each *Client Set* and similarity across different clusters to achieve better global model accuracy. For example, as Fig. 1 shows, clients are arranged to each FL subgroup to maximize the similarity between each edge server and the *Virtual Target*. In the experiments, our evaluation results justify the superior performance of VISIT in mitigating non-IID conditions compared to existing HFL frameworks, achieving a maximum improvement of 41% accuracy and reducing total communication rounds to 82%. We also observed an insightful phenomenon that VISIT arranging each FL subgroup with a small number of clients strikes a good balance between global model performance and communication costs.

II. System Model

This section first provides HFL architecture, which delineates a client-edge-cloud hierarchy that distributes learning across multiple computational layers. This indicates scalability and efficiency benefits that typical centralized FL cannot accomplish. Subsequently, we advocate sequential training as a promising approach to address non-IID by balancing the accumulation of all clients' data distribution within each sub-FL group.

A. Hierarchical Federated Learning

The objective of the FL framework is to acquire a global model parameterized by w by utilizing data distributed across N clients while ensuring each local data's privacy. Every device, denoted as $n \in [N]$, can access samples from its local dataset $\mathcal{D}_n$. The loss function evaluates the variation between the model's prediction and the actual value for the m-th data sample, represented as $f_m(w)$.

The Federated Averaging (FedAvg) algorithm [2] adopts an iterative methodology to minimize the overall loss function $F(w)$, which can be derived as a weighted average of the local loss functions $F_n(w)$ on local datasets $\mathcal{D}_n$. $F(w)$ and $F_n(w)$ are specified as follows:

$$F(\boldsymbol{w}) = \frac{\sum_{n=1}^{N} |\mathcal{D}_n| F_n(\boldsymbol{w})}{|\mathcal{D}|}, \quad F_n(\boldsymbol{w}) = \frac{\sum_{m \in \mathcal{D}_n} f_m(\boldsymbol{w})}{|\mathcal{D}_n|}.$$

To accommodate more clients at a significantly reduced cost of communication, we consider the HFL system [4], which consists of one cloud server, L edge servers indexed by ℓ, with disjoint *Client Set* $\{C^\ell\}_{\ell=1}^{L}$, and N clients indexed by n. Let $\mathcal{D}^\ell$ represent the aggregated dataset under edge ℓ. Alone in the HFL system, following every i local update performed on each client, each edge server conducts the aggregation for the models of its respective clients. After each iteration of j edge model aggregations, the cloud server aggregates the models from all edge servers. This implies that communication with the cloud occurs every ij local update. $w_n^\ell(k)$ represents the local model parameters after the k-th local update. The progression of local model parameters $w_n^\ell(k)$ can be described as follows:

$$w_n^\ell(k) = \begin{cases} w_n^\ell(k-1) - \nabla F_n(w_n^\ell(k-1)) & k|i \neq 0 \\ \frac{\Sigma_{n \in C^\ell} |D_n| [w_n^\ell(k-1) - \nabla F_n(w_n^\ell(k-1))]}{|D^\ell|} & k|ij \neq 0 \\ & k|i = 0 \\ \frac{\Sigma_{n=1}^{N} |D_n| [w_n^\ell(k-1) - \nabla F_n(w_n^\ell(k-1))]}{|D|} & k|ij = 0 \end{cases}$$

B. Sequential Training

In a realistic scenario, we cannot guarantee that individual datasets obtained from different clients were sampled independently from an identical underneath distribution: $\mathcal{P}(\mathcal{D}_x) \neq \mathcal{P}(\mathcal{D}_y)$ for every pair of clients x and y. Therefore, we advocate sequential training [13] to address the non-IID issue by reorganizing the accumulated clients' data distribution among *Client Set* within each edge server. That is, the disjoint *Client Set* C^ℓ of each edge server consists of clients observing distinct data; thereby, each set should approximate the identical underneath distribution among every κ class, $, i.e., \bigcup_{n \in C^\ell} \mathcal{D}_n \sim \mathcal{U}_{[\kappa]} \forall \ell$. Through conducting sequential training instead of the conventional FedAvg under each edge server, local models can intuitively gather information regarding the more significant number of classes, even in cases where individual clients show substantial heterogeneity. Previous research hasn't attempted integrating HFL with sequential training. The adoption of HFL addresses the performance constraint experienced by a single server. In fact, a well-arranged *Client Set* combined

with sequential training allows the local model to accumulate a diverse and balanced data distribution during the training process, which in turn mitigates the impact of non-IID data. As a result, it is essential to have a diverse but balanced *Client Set* to achieve quick and effective convergence to high accuracy in the HFL system under non-IID conditions. Once all *Client Set* include a wide variety of clients, the similarity between D^{ℓ} will increase, representing the degree of non-iid issue mitigated at the edge-cloud layer.

C. Design Objectives

In general, applying *Client Set* with lower Intra-CS and higher Inter-CS can accelerate convergence and improve the accuracy of HFL systems under non-IID situations. A lower Intra-CS implies greater variety under an edge server. In contrast, a higher Inter-CS indicates a reduced level of non-IID across edge servers. However, merely minimizing Intra-CS may result in a favor to the outlier clients instead. Furthermore, maximizing Inter-CS between every *Client Set* consisting of similar outlier clients would yield incomprehensive global model training results. Therefore, we design a *Virtual Target* by approximating a uniformly distributed dataset for organizing the *Client Set*. In fact, this target could infer the performance of the model obtained through centralized learning, which enables us to minimize the Intra-CS and maximize the Inter-CS by maximizing the average similarity between each *Client Set* and the *Virtual Target* (VTS). This indicates that the *Client Set* composition must be diverse and balanced, with no preference given to any particular client; otherwise, the VTS will decrease.

In our target framework, grouping clients with diverse data distributions can enhance the performance of each global model and improve its resilience against non-IID with sequential training [13]. Intuitively, increasing the number of clients assigned to each edge server can improve its resilience against non-IID challenges. However, this inevitably leads to longer sequential training durations because training processes cannot be parallelized with more clients in the same FL subgroup [14]. Therefore, to mitigate the impact of sequential training on the effectiveness of FL training, it is challenging to propose a *Client Set* arranging strategy that maintains model performance, communication costs, and non-IID issues jointly. Accordingly, the design objectives for our proposed system are as follows:

1) Privacy Conserving: To obtain *Client Set*, the system must estimate the client's distribution while following FL privacy standards. Consequently, the system must include client estimation techniques, including estimating the *Virtual Target* above, without compromising privacy.

2) Balancing Diversity in Client Set*:* We intend to establish a *Virtual Target* for arranging each *Client Set* to improve the accuracy of the global model. The primary goal is to maximize the model's accuracy by maximizing the similarity between each *Client Set* and the *Virtual Target*.

3) Limited Number of Clients per Edge: The proposed system should sustain its efficacy in addressing non-IID scenarios and minimize the adverse consequences of sequential training while arranging fewer clients at an edge.

III. VIRTUAL-TARGETED SEQUENTIAL TRAINING WITH HIERARCHICAL FEDERATED LEARNING

In this section, we present **VI**rtual-targeted **S**equent**I**al **T**raining with Hierarchical Federated Learning (*VISIT*), a framework for organizing *Client Set* to achieve balanced data distributions while upholding privacy. We aim to accumulate diverse clients inside the local model of each FL subgroup through the sequential training phase, mitigating the non-IID impact. The objective is to obtain the *Client Set* that maximizes the similarity between different *Client Set* and minimizes it inside each individual *Client Set* by maximizing the similarity between *Client Set* and *Virtual Target*. The procedure involves four main steps: estimating clients, defining arranging indicators, formulating *Virtual Target*, and arranging *Client Set* without accessing client data. First, we design a novel approach to estimating clients' data distributions using secure *signature vectors* derived from local models tested on public data. After that, we develop an arranging strategy that harnesses these *signature vectors* in conjunction with a *Virtual Target* to organize each *Client Set* systematically.

A. Estimating Clients

The approach employed in [8], [10], [11] is applied in this study, where locally trained models are utilized to estimate the distribution of the clients' data. We employ a pre-training phase for predicting the distribution of each client's local dataset, in which every individual client, denoted as n, generates a model θ_{w_n} with an output layer using the softmax function by conducting training on its respective local dataset for a specified number of epochs, e, commencing with an identical random initialization point, w_0. We then present an approach that leverages local models $\theta_{w_n} n \in [N]$ to devise effective strategies following the pre-training phase. Such an approach is formulated based on the predictions obtained from the individual client's local model θ_{w_n} on a publicly available dataset $\mathcal{D}_{pub} = \bigcup_{c=1}^{N_c} \mathcal{D}_c$, in which $\mathcal{D}_c$ comprises M_c data for class $c \in [\kappa]$, hosted on the server. After conducting tests on each local model θ_{w_n} with $\mathcal{D}_{pub}$, we calculate the average prediction score of each local model for every data point $\{x_i, y_i\}_{i=1}^{M_c} = \mathcal{D}_c$ on the corresponding label c, which can be expressed as $v_{n,c} = \frac{1}{M_c} \sum_{x \in \mathcal{D}_c} \theta_{w_n}(x)$. After all, we could describe the n-th client's *signature vector* as:

$$v_n := [v_{n,1}, \ldots, v_{n,\kappa}] \in [0,1]^{\kappa}$$

Given that the predictions of the n-th model are biased into the greater number of classes observed in $\mathcal{D}_n$ [15], it can be concluded that v_n serves as a suitable secure illustration of $\mathcal{D}_n$.

B. Arranging Indicator

Next, we use the client n's estimation v_n to construct similar *Client Set* among clients with various distributions. To achieve this goal, assuming we have v_i and v_j, an indicator $\iota(v_i, v_j)$ must be devised for evaluating the degree of similarity between the two given distribution estimations. In this paper, we adopt *cosine similarity* as the metric for measuring *signature vector* of the clients' predictions; however, other indicators are

Algorithm 1 Arranging *Client Set*

Input: $N, L, k, v^{\text{target}}$

1: **function** ESTIMATESIGNATURES($N, \mathcal{D}_{\text{pub}}, e$)
2: **for** $n \leftarrow 1$ **to** N **do**
3: $\theta_{w_n} \leftarrow$ train model on $\mathcal{D}_n$ for e epochs from w_0
4: **for** $c \leftarrow 1$ **to** κ **do**
5: $v_{n,c} \leftarrow \frac{1}{M_c} \sum_{x \in \mathcal{D}_c} \theta_{w_n}(x)$
6: **end for**
7: $v_n \leftarrow [v_{n,1}, \ldots, v_{n,\kappa}]$
8: **end for**
9: **return** $\{v_n\}_{n=1}^N$
10: **end function**
11: $\{v_n\}_{n=1}^N \leftarrow$EstimateSignatures($N, \mathcal{D}_{\text{pub}}, e$)
12: **for** $\ell \leftarrow 1$ **to** L **do**
13: $C^\ell \leftarrow \emptyset$
14: $C^\ell \leftarrow C^\ell \cup \{i\}$ where i is chosen randomly from $[N]$
15: $v^{C^\ell} \leftarrow \frac{1}{|C^\ell|} \sum_{n \in C^\ell} v_n$
16: **for** $j \leftarrow 1$ **to** $k-1$ **do**
17: $\tilde{C}^\ell \leftarrow C^\ell \cup \{j\}$ where $j = \arg\max_{j \in [N]} \iota(v^{\tilde{C}^\ell}, v^{\text{target}})$
18: $C^\ell \leftarrow \tilde{C}^\ell$
19: $v^{C^\ell} \leftarrow \frac{1}{|\tilde{C}^\ell|} \sum_{n \in \tilde{C}^\ell} v_n$
20: **end for**
21: **end for**
22: **return** $\{C^\ell\}_{\ell=1}^L$

available. Meanwhile, more advanced methodologies, including capturing feature representations and cluster assignment, could further take into consideration [16]. Due to the inherent privacy-preserving nature of FL, getting the precise amount of data held by any individual client is forbidden. Therefore, we advocate an angle-oriented strategy to derive angular separation between two vectors rather than the magnitude of the vectors.

C. *Formulating* Virtual Target

To achieve a higher degree of similarity across the balance *Client Set*, an overall aim is needed for assigning each set during the arranging phase. Nevertheless, due to the inability to gather data from all clients to get comprehensive training data, we expand the concept of the *signature vector* from the client estimation phase by using the same approach to estimate the overall aim's *signature vector*, called *Virtual Target* v^{target}. In brief, our goal is to get a higher prediction score for all labels inside each set of clients, which means that the overall aim's *signature vector* can be described as $v^{target} := [1, \ldots, 1] \in [1]^\kappa$. Specifically, maximizing the similarity between a *Client Set* and v^{target} indicates that this *Client Set* consists of diverse, well-balanced clients with data of varying labels. When we set up such a unified v^{target} as a target for arranging each *Client Set*, the similarity between the *Client Set* will be maximized.

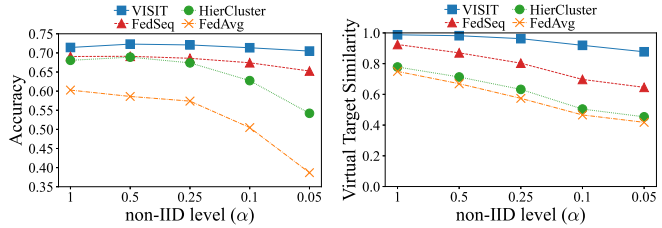
D. *Arranging Client Set*

Initially, we introduce the accumulated *Client Set signature vector*, which consists of clients' *signature vector* v_n, and those clients belong to the *Client Set* C^ℓ. This accumulated *signature vector* could be described as $v^{C^\ell} = \frac{1}{|C^\ell|} \sum_{n \in C^\ell} v_n$. Given L edge servers, each of which can connect with k clients, our objective is to find members of each edge server that maximize the similarity between v^{C^ℓ} and v^{target}. This entails finding L *Client Set*, each containing k clients. In Algorithm 1, we show how to approximate the maximization problem by utilizing the *signature vector* of each client and the *Virtual Target*. We start by arranging a client $i \in [N]$, which is randomly chosen, to the current *Client Set*. Afterward, we could derive the *signature vector* of the current *Client Set* v^{C^ℓ} (line 15). Then choose the next client j that could take the v^{C^ℓ} closest to the v^{target} (maximizing VTS), *i.e.* $\max_{j \in [N]} \iota\left(v^{\tilde{C}^\ell}, v^{target}\right), \tilde{C}^\ell = C^\ell \cup j$ (line 17). Repeating the processes above for each *Client Set* until there are k clients in the L *Client Set* while continuously maximizing $\iota\left(\frac{1}{|C^\ell|+1} \sum_{n \in C^\ell \cup j} v_n, v^{target}\right)$. The behavior of putting maximizing VTS first makes it tend to choose less diverse clients for balance, even if more diverse clients exist.

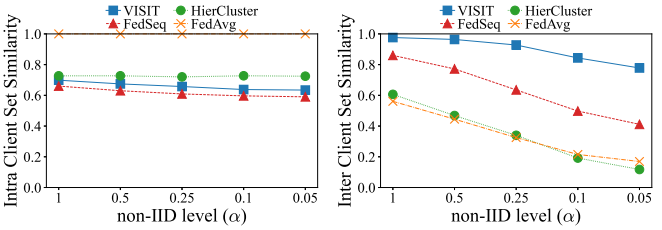
IV. PERFORMANCE EVALUATION

A. *Simulation Settings*

We conducted the experiments using two widely used datasets, EMNIST and CIFAR-10. Both datasets are trained using the convolutional neural network (CNN). We use the Dirichlet distribution [17], a continuous multivariate probability distribution, to test different non-IID scenarios. We evaluate different non-IID scenarios with $\alpha \in \{1, 0.5, 0.25, 0.1, 0.05\}$. (As the parameter α decreases, the nodes' data distributions become increasingly non-IID). Three distinct model similarities can be calculated using the arranging indicator ι to evaluate the degree to which the non-IID problem is alleviated. 1) Intra-CS: The mean similarity between two clients within each *Client Set*, denoted as $\frac{1}{Lk} \sum_{\ell \in [L]} \sum_{\{i,j\} \in C^\ell} \iota(v_i, v_j)$, 2) Inter-CS: The mean similarity between each set of clients, denoted as $\frac{1}{L} \sum_{\{i,j\} \in [L]} \iota(v^{C^i}, v^{C^j})$, and 3) VTS: The mean similarity between each set of clients and the v^{target}, denoted as $\frac{1}{L} \sum_{\ell \in [L]} \iota(v^{C^\ell}, v^{target})$. We compare VISIT with the conventional FedAvg [2], HierCluster [11] and FedSeq [13]. As previously shown, HierCluster and FedSeq didn't consider the HFL design. Therefore, in the subsequent simulations, we apply the same HFL architecture and sequential training strategy for HierCluster and FedSeq without modifying their fundamental principles. The key idea of HierCluster involves the grouping of similar clients together. For HierCluster, hierarchical clustering was used to split the clientele into a total of L clusters; inside each *Client Set*, the clients were arranged based on their respective cluster membership. For FedSeq, the authors suggest choosing clients to arrange into the *Client Set* based on their dissimilarity to the current *Client Set*. Two metrics are adopted to evaluate performance: 1) test accuracy and 2) *Client Set* similarity metrics. Considering *Client Set* similarity metrics, a lower value of Intra-CS indicates a higher level of diversity among the clients included in the *Client Set*. Conversely, a higher value of Inter-CS suggests a lesser degree of non-IID issues between the edge servers in the middle tier. Lastly, a higher value of VTS signifies a more comprehensive representation of class information for each *Client Set*. Note that each result is averaged over ten trials. For the EMNIST dataset, the chosen target accuracy, denoted as ζ, is specified at 60% ($\zeta = 50\%$ for CIFAR-10).

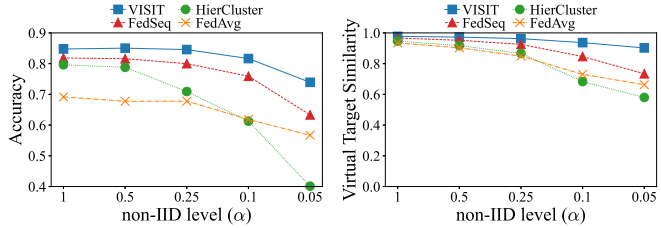


(a) Accuracy (b) Virtual Target Similarity

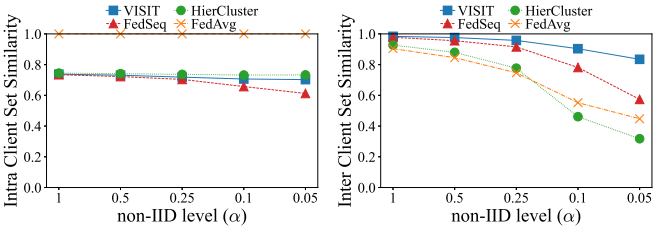


(c) Intra Client Set Similarity (d) Inter Client Set Similarity

Fig. 2: Effect of different level of non-IID data on accuracy and similarity (CIFAR-10).

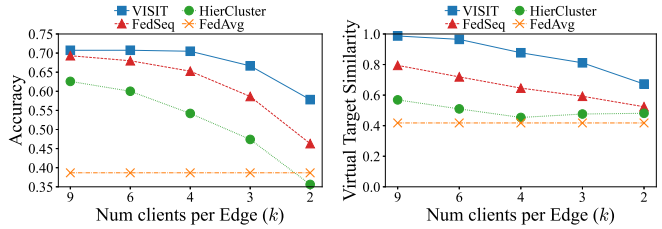


(a) Accuracy (b) Virtual Target Similarity

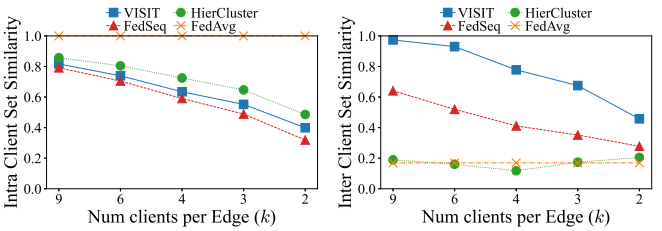


(c) Intra Client Set Similarity (d) Inter Client Set Similarity

Fig. 4: Effect of different level of non-IID data on accuracy and similarity (EMNIST).

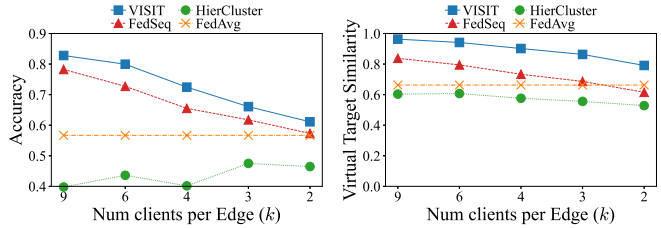


(a) Accuracy (b) Virtual Target Similarity

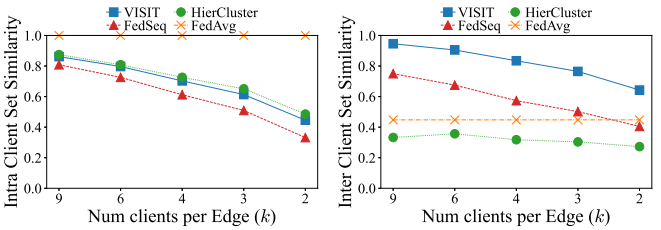


(c) Intra Client Set Similarity (d) Inter Client Set Similarity

Fig. 3: Effect of different number of client arranged to a edge server on accuracy and similarity (CIFAR-10).



(a) Accuracy (b) Virtual Target Similarity



(c) Intra Client Set Similarity (d) Inter Client Set Similarity

Fig. 5: Effect of different number of client arranged to a edge server on accuracy and similarity (EMNIST).

B. Effects of non-IID Level on Client Set *Similarity and Model Performance*

We conduct a comparative analysis of each approach, testing various data distributions within the context of the HFL framework of $\{L, k\} = \{9, 4\}$. About the FedAvg, we perform a standard FL configuration, selecting $L \times k$ clients for each round. Fig. 2 and Fig. 4 show that, in various non-IID situations, VISIT exhibits an improved ability to decrease the Intra-CS and raise the Inter-CS by maintaining a higher VTS compared to other methods. The concept of FedSeq enables it to achieve the lowest level of Intra-CS. However, its performance in terms of Inter-CS is less effective than VISIT. On the other hand, HierCluster succeeds in grouping similar clients within the same *Client Set*, leading to the highest Intra-CS. Consequently, this results in a lower Inter-CS. Notably, when the non-IID situation is severe, HierCluster's Inter-CS may be less than FedAvg impacting the VTS. Lastly, FedAvg does not employ sequential training. As a result, the $L \times k$ clients chosen in each round are treated as $L \times k$ *Client Set*, each containing only one member. Furthermore, since FedAvg does not utilize an arbitrary arranging method to determine the selection of these *Client Set*, it generally exhibits lower values for Inter-CS and VTS compared to other approaches. In contrast, both FedSeq and VISIT obtained high Inter-CS and VTS, VISIT performed even better than FedSeq, particularly in CIFAR-10, as demonstrated in Fig. 2(b) and Fig. 4(b). The reason is that using CIFAR-10, which has only 10 classes, causes clients to have fewer classes on average, and VISIT is more likely to select clients with more classes to balance the *Client Set*. In contrast, FedSeq will select clients with fewer

TABLE I: Improvement in Accuracy

Method	non-IID	EMNIST (200 rounds)	CIFAR-10 (300 rounds)
FedAvg	$\alpha = 0.1$	61.851% (1.0x)	50.470% (1.0x)
HierCluster		61.271% (0.991x)	62.749% (1.243x)
FedSeq		75.888% (1.226x)	67.419% (1.335x)
VISIT		**81.650% (1.320x)**	**71.348% (1.414x)**

TABLE II: Rounds to Achieve the Target Accuracy

Method	non-IID	EMNIST ($\zeta = 60\%$)	CIFAR-10 ($\zeta = 50\%$)
FedAvg	$\alpha = 0.1$	132 (1.0x)	284 (1.0x)
HierCluster		140 (1.06x)	107 (0.38x)
FedSeq		60 (0.45x)	63 (0.22x)
VISIT		**32 (0.24x)**	**51 (0.18x)**

classes by choosing the least similar clients. Overall, it can be indicated with Fig. 2(a) and Fig. 4(a) that the accuracy of the approaches above fluctuates with the variations in similarities.

As Table I and Table II show, while VISIT mitigates the non-IID problem as well as the other methods, it pays extra attention to the balancing of *Client Set*, thus having *Client Set* capture a more complete distribution of clients for converging faster and improving accuracy. In Fig. 2(c) and Fig. 4(c), FedSeq prioritizes the objective of minimizing Intra-CS, leading to a relatively smaller impact than the other similarities. Although VISIT has slightly lower performance than FedSeq in terms of Intra-CS, it demonstrates an increased level of dominance in both Inter-CS and VTS. Such result shows that FedSeq tends to optimize Intra-CS, which triggers additional bias towards selecting a specific least-similar set of clients, thus sacrificing Inter-CS without arranging a well-balanced *Client Set*.

C. *Effect of Number of Clients per Edge Server on* Client Set *Similarity and Model Performance*

We examine different combinations for the capacity of clients that can be accommodated by each edge server, i.e., $\{L, k\} \in \{\{4, 9\}, \{6, 6\}, \{9, 4\}, \{12, 3\}, \{18, 2\}\}$. Meanwhile, we set α to 0.05 for every combination. Specifically, the performance of FedAvg remains constant regardless of the value of k due to using parameters equivalent to $\{L, k\} = \{36, 1\}$. Fig. 3 and Fig. 5 show that when the k value is reduced, VISIT consistently exhibits superior accuracy compared to other approaches. In Figs. 3(c) and 5(c), although FedSeq outperforms other methods in Intra-CS, VISIT's accuracy still outperforms FedSeq, indicating that pursuing diversity in each FL subgroup is insufficient. Besides, the lower accuracy of HierCluster, especially on EMNIST dataset (with more categories) compared to FedAvg, is primarily because the richer client distribution makes the clusters distinguished by hierarchical clustering more distinct from one another, resulting in each edge server being more specialized in their particular model. Therefore, HierCluster yields poorer outcomes on Inter-CS and VTS than FedAvg, as shown in Figs. 5(b) and 5(d). As for FedSeq, when k is relatively large, the bias towards a minority of clients will be eased, and the performance will be comparable to that of VISIT. However, when k is lower, FedSeq fails to attain balance by consistently selecting the least similar clients. In contrast, VISIT can arrange balanced *Client Set* by approaching v^{target} as the first priority, slightly sacrificing Intra-CS to select clients with more classes to achieve higher VTS, and wisely select clients with fewer classes if k is large enough to gain diversity.

V. Conclusion

In this paper, we propose **VI**rtual-targeted **S**equent**I**al **T**raining with HFL (VISIT) framework to address existing FL research issues. However, VISIT faces design challenges such as preserving privacy during distribution estimation and balancing *Client Set* diversity without additional communication costs. Therefore, we introduce *Virtual Target Similarity* (VTS) metric to arrange *Client Set* by maximizing the average similarity between each *Client Set* and the *Virtual target*. This strategy indicates that the composition of the *Client Set* among FL subgroups must be diverse and balanced jointly. The performance evaluation on EMNIST and CIFAR-10 datasets shows that VISIT can reduce training rounds by 82% and improve model accuracy by 41% compared to other state-of-the-art baselines. Specifically, the results emphasize the negative effect that unbalanced *Client Set* exhibits on the VTS, which in turn lowers model performance. Instead, VISIT slightly compromises Intra-CS for higher Inter-CS and VTS indicates its success in harmonizing the client distribution of FL subgroups to improve global model performance with non-IID data. Prioritizing maximum VTS allows VISIT to lessen non-IID impact under fewer clients per *Client Set*, which strikes a balanced tradeoff between model performance and communication cost.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[2] B. McMahan *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Proc. of AISTATS*, 2017.

[3] Z. Zhong, Y. Zhou, D. Wu, X. Chen, M. Chen, C. Li, and Q. Z. Sheng, "P-fedavg: Parallelizing federated learning with theoretical guarantees," in *Proc. of IEEE INFOCOM*, 2021.

[4] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. of IEEE ICC*, 2020.

[5] N. Mhaisen, A. A. Abdellatif, A. Mohamed, A. Erbad, and M. Guizani, "Optimal user-edge assignment in hierarchical federated learning based on statistical properties and network topology constraints," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 55–66, 2021.

[6] D. Caldarola, M. Mancini, F. Galasso, M. Ciccone, E. Rodolà, and B. Caputo, "Cluster-driven graph federated learning over multiple domains," in *Proc. of CVPR*, 2021.

[7] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," in *Proc. of NeurIPS*, 2020.

[8] G. Long *et al.*, "Multi-center federated learning: clients clustering for better personalization," *World Wide Web*, vol. 26, pp. 481–500, 2023.

[9] K. Kopparapu and E. Lin, "Fedfmc: Sequential efficient federated learning on non-iid data," *arXiv preprint arXiv:2006.10937*, 2020.

[10] F. Sattler *et al.*, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, pp. 3710–3722, 2020.

[11] C. Briggs *et al.*, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *Proc. of IJCNN*, 2020.

[12] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *Proc. of IEEE INFOCOM*, 2020.

[13] R. Zaccone, A. Rizzardi, D. Caldarola, M. Ciccone, and B. Caputo, "Speeding up heterogeneous federated learning with sequentially trained superclients," in *Proc. of IEEE ICPR*, 2022.

[14] M. Gharibi, S. Bhagavan, and P. Rao, "Federatedtree: A secure serverless algorithm for federated learning to reduce data leakage," in *Proc. of IEEE Big Data*, 2021.

[15] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. on Knowl. Data. Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.

[16] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[17] S. P. Sturluson *et al.*, "Fedrad: Federated robust adaptive distillation," *arXiv preprint arXiv:2112.01405*, 2021.