arXiv:2505.18866v1 [cs.LG] 24 May 2025

Distribution-Aware Mobility-Assisted Decentralized Federated Learning

Md Farhamdur Reza[1], Reza Jahani[1], Richeng Jin[2], and Huaiyu Dai[1]
[1]NC State University [2]Zhejiang University
{mreza2, rjahani, hdai}@ncsu.edu richengjin@zju.edu.cn

***Abstract*—Decentralized federated learning (DFL) has attracted significant attention due to its scalability and independence from a central server. In practice, some participating clients can be mobile, yet the impact of user mobility on DFL performance remains largely unexplored, despite its potential to facilitate communication and model convergence. In this work, we demonstrate that introducing a small fraction of mobile clients, even with random movement, can significantly improve the accuracy of DFL by facilitating information flow. To further enhance performance, we propose novel distribution-aware mobility patterns, where mobile clients strategically navigate the network, leveraging knowledge of data distributions and static client locations. The proposed moving strategies mitigate the impact of data heterogeneity and boost learning convergence. Extensive experiments validate the effectiveness of induced mobility in DFL and demonstrate the superiority of our proposed mobility patterns over random movement.**

***Index Terms*—Mobility, Distribution-Aware, Decentralized Federated Learning, Clustering.**

I. Introduction

The rapid growth of machine learning applications across diverse domains has driven increasing demand for efficient and scalable training frameworks. Traditional centralized frameworks rely on clients sending raw data to a central server for model training. However, this approach has significant drawbacks, including high network traffic and privacy concerns, as raw personal data is transmitted over the network [1].

Federated Learning (FL) [2] enables model training without sharing raw data, with each client training a local model and sending only updates to a central server for aggregation. However, FL faces several challenges, particularly data heterogeneity, where client data is often not independently and identically distributed (non-IID) in real-world scenarios. This heterogeneity leads to performance degradation, where conventional aggregation methods fail to generalize well across clients [3]. To address this issue, various strategies have been proposed. For instance, FedAF [4] improves aggregation by adapting weights based on data distribution, outperforming standard approaches like FedAvg [2] and FedProx [5]. Similarly, FedDisco [6] introduces a distribution-aware strategy to address class imbalance across clients. Another challenge is communication overhead, which grows significantly with more clients, leading to scalability concerns [7].

Decentralized Federated Learning (DFL) extends FL by eliminating the need for a global server, utilizing peer-to-peer communication for model aggregation [8]. In DFL, clients exchange model updates directly with their neighbors, defined by their communication range. Although this decentralized setup improves scalability and eliminates the reliance on a central server, it introduces new challenges.

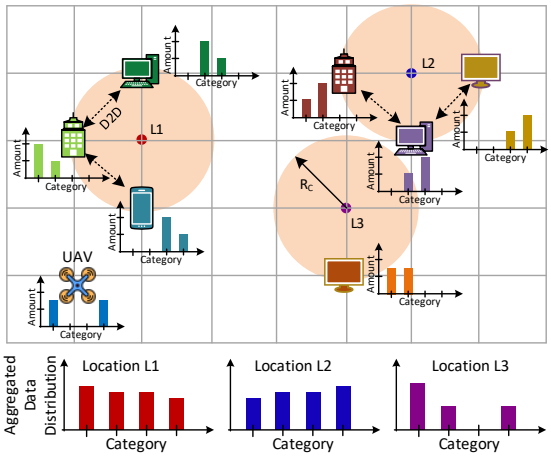


Fig. 1: DFL schematic with a UAV as a mobile client and seven fixed clients. It illustrates the local data distributions of each client and the UAV's movement across three locations (L1, L2, and L3). The shaded regions represent communication ranges, and the aggregated data distribution at each UAV location is shown at the bottom for reference.

One critical factor affecting DFL performance is network topology [9], [10]. Given a communication radius, a client participating in DFL is restricted to communicating with neighbors within the communication range. The structure of the resulting communication graph significantly influences information flow, model convergence, and overall performance. In sparse network topologies, limited connectivity reduces the flow of updates, leading to slower convergence than denser topologies. Additionally, information may fail to propagate across disconnected networks, further hindering convergence.

In many real-world applications, clients are not always static; some exhibit mobility [11], which influences communication patterns and information exchange in decentralized networks. Motivated by this, we investigate the impact of mobility in DFL to understand its effects on model performance and information flow. Mobile clients move freely within the network following a predefined mobility pattern, while static clients remain fixed. We first explore the impact of mobility by considering random movement and demonstrate that introducing mobile clients enhances performance

by improving information dissemination across the network. These mobile clients explore the network, gather updates from fixed clients, and propagate information to other disconnected areas, mitigating the adverse effects of network sparsity.

To further utilize the benefits of mobility, we propose distribution-aware mobility patterns to enhance convergence by addressing both network sparsity and data heterogeneity. In this approach, mobile clients move based on knowledge of the data distributions and locations of fixed clients. Their trajectories are guided by an aggregated data distribution metric that determines effective movement paths to reduce heterogeneity and improve overall performance. A *simple illustration* of the proposed mobility pattern is provided in Fig. 1, where an unmanned aerial vehicle (UAV) acts as a mobile client which is allowed to move across three locations: L1, L2, and L3. If L1 is the current location of the UAV, with the proposed approach, the UAV is most likely to move towards L3 to reduce the impact of data heterogeneity and sparsity. Our study further examines the impact of network connectivity parameters, such as communication radius, number of mobile clients, and their speed, on DFL performance.

To the best of our knowledge, this is one of the first works to investigate the impact of mobility in the DFL setting considering different networking parameters (such as communication radius, number of mobile clients, and velocity of mobile clients) and to design mobility patterns accordingly for performance improvement. Our main contributions are summarized as follows:

- We demonstrate that mobility enhances performance in decentralized federated learning.
- We propose two distribution-aware mobility patterns, which outperform random movement and mitigate the issues caused by data heterogeneity.
- We conduct extensive experiments to reveal the impact of various networking parameters on overall model performance.

II. Related Works

Comprehensive convergence analyses for DFL setting are provided in [12], [13]. The positive correlation between the convergence rate and the spectral gap of the communication topology in decentralized settings is highlighted in [9]. To address the slow convergence rate, the authors in [14] propose Gossip SGD for distributed learning. Moreover, a mutual knowledge transfer technique to mitigate performance degradation in DFL systems is introduced in [15]. The authors in [16] demonstrate that a carefully designed topology can reduce the impact of data heterogeneity. Meanwhile, a novel framework for mixing matrix optimization, aimed at minimizing gradient mixing error in decentralized settings, is proposed in [17]. Similarly, the authors in [18] consider the joint impact of topology and data heterogeneity on the convergence behavior of decentralized learning.

Existing work on the DFL setting has largely overlooked the impact of user mobility, despite its potential to significantly influence system performance. Several studies explore user mobility in the classic FL context [11], [19]–[22]. Model aggregation strategies aimed at optimizing federated learning performance in vehicular networks are proposed in [11], [19]. The authors in [20] investigate the impact of user mobility in classic FL and identify it as a negative factor causing performance degradation. In contrast, it is suggested in [21] that mobility can be beneficial when the users move at an appropriate speed. The advantage of mobility in asynchronous FL is investigated in [22]. However, none of these studies explore the effect of user mobility in fully decentralized settings. Although existing work has explored the use of UAVs in the DFL setting, they often overlook the impact of mobility [23]. In contrast, the author of [24] explores user mobility in DFL, but only in terms of random movements.

III. System Model

We consider a decentralized network of C clients denoted by the set $\mathcal{C} = [C] := \{1, ..., C\}$. Among these, a subset $\mathcal{C}_m \subset \mathcal{C}$ comprises mobile clients that move according to a predefined mobility pattern, while the remaining clients, denoted as $\mathcal{C}_s = \mathcal{C} \setminus \mathcal{C}_m$, remain static throughout the process. Each client $i \in \mathcal{C}$ holds a local dataset $\mathcal{D}_i = \{(\boldsymbol{s}_n, y_n) \mid n = 1, 2, ..., |\mathcal{D}_i|\}$ and updates its local model x_i based on its local dataset, where $\boldsymbol{s}_n$ and y_n denote the n-th input and corresponding true label. DFL objective is to optimize the following function [12]:

$$f^* := \Big[\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} f_i(x) \Big], \tag{1}$$

where $f_i(x) := \mathbb{E}_{\xi_i \sim \mathcal{D}_i} F_i(x; \xi_i)$ represents the local expected loss over mini-batches ξ_i, with $F_i(.)$ denoting the client-specific loss function, and d being the model dimensionality.

Each client i has a limited communication range defined by a circular coverage area of radius R_c. A client i can communicate with another client $j \in \mathcal{C} \setminus \{i\}$ if j lies within this coverage. The set of connected neighbors for client i at global round t is given by: $\mathcal{N}_i^{(t)} = \{j \mid \|L_i^{(t)} - L_j^{(t)}\|_2 \leq R_c, \forall j \in \mathcal{C} \setminus \{i\}\}$, where $L_i^{(t)}$ represents the location of client i at the global round t. While fixed clients remain at the same locations, i.e., $L_i^{(t-1)} = L_i^{(t)}$, $\forall i \in \mathcal{C}_s$, mobile clients change their locations according to a mobility pattern constrained by the maximum movable radius R_m. Consequently, the network topology varies over time due to mobile clients.

In this decentralized framework, each client maintains local parameters $x_i^{(t-1)} \in \mathbb{R}^d$ and computes the local stochastic gradient $g_i^{(t-1)} := \nabla F_i(x_i^{(t-1)}, \xi_i)$ based on samples from its dataset at round t. Following this, each client $i \in \mathcal{C}$ exchanges models with its neighbors $\mathcal{N}_i^{(t)}$ through D2D communication and aggregates the received models. The decentralized training process involves two phases per round:

1) *Local update*: $x_i^{(t-\frac{1}{2})} = x_i^{(t-1)} - \eta g_i^{(t-1)}$, where η is the learning rate.
2) *Consensus update*: $x_i^{(t)} = \sum_{j=1}^{N} w_{i,j}^{(t)} x_j^{t-\frac{1}{2}}$, where $w_{i,j}^{(t)}$ is the (i,j)-th entry of the mixing matrix $\boldsymbol{W}^{(t)} \in \mathbb{R}^{C \times C}$, satisfying $w_{i,j}^{(t)} = 0$ if $j \notin \mathcal{N}_i^{(t)} \cup \{i\}$.

For convenience, we use the following matrix notation to stack all the models, $\boldsymbol{X}^{(t)} = \left[x_1^{(t)}, ..., x_C^{(t)}\right] \in \mathbb{R}^{d\times C}$. Likewise, we define $\boldsymbol{G}^{(t)} = [g_1^{(t)}, ..., g_C^{(t)}]$. Hence, the local and consensus updates for all the models can be expressed as: $\boldsymbol{X}^{(t-\frac{1}{2})} = \boldsymbol{X}^{(t-1)} - \eta \boldsymbol{G}^{(t-1)}$ and $\boldsymbol{X}^{t} = \boldsymbol{X}^{(t-\frac{1}{2})}\boldsymbol{W}^{(t)}$, respectively, where $\boldsymbol{W}^{(t)}$ is the time-varying mixing matrix due to the mobile clients. Non-zero weights of the consensus matrix $\boldsymbol{W}^{(t)}$ are determined based on the local dataset sizes of neighboring clients, following FedAvg [2]. The (i,j)-th entry of $\boldsymbol{W}^{(t)}$ is as follows:

$$w_{i,j}^{(t)} = \begin{cases} \frac{|\mathcal{D}_j|}{|\mathcal{D}_i| + \sum_{k\in\mathcal{N}_i^{(t)}} |\mathcal{D}_k|}, & \text{if } j \in \mathcal{N}_i^{(t)} \cup \{i\}, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

IV. Mobility Models

In the mobility model, we consider clients positioned on a $G \times G$ grid, where each location is represented as $(p,q) \in \mathcal{G}_{\mathcal{L}}$ with $\mathcal{G}_{\mathcal{L}} = \{(p,q) \mid p,q \in \{1,2,...,G\}\}$. At the beginning of the training process, each user i is randomly assigned an initial location $L_i^{(0)} \in \mathcal{G}_{\mathcal{L}}$ within the grid. Based on the mobility pattern, each mobile client in $\mathcal{C}_m$ relocates to a new grid position in each global round, and all clients in $\mathcal{C}$ exchange their models with their neighboring clients for aggregation. In this section, we discuss baseline mobility models and elaborate on the proposed distribution-aware mobility patterns.

A. Baseline Mobility Patterns

We consider two baseline models: *Static* and *Random Movement*. For the **Static** scenario, client locations remain fixed throughout the training, i.e., $L_i^{(t-1)} = L_i^{(t)}; \forall i \in \mathcal{C}, \forall t \in [T]$, where T is the total number of global training rounds. For the **Random Movement**, mobile clients $i_m \in \mathcal{C}_m$ uniformly randomly select a new location $L_{i_m}^{(t)}$ within the grid at each round t, subject to their location at the end of round $t-1$ and a maximum allowable displacement per round R_m. This constraint ensures that clients move within a feasible range per round, maintaining realistic mobility patterns. Formally,

$$L_{i_m}^{(t)} \sim \text{Uniform}\Big(\{L \mid \|L - L_{i_m}^{(t-1)}\|_2 \le R_m, L \in \mathcal{G}_{\mathcal{L}}\}\Big). \tag{3}$$

B. Proposed Mobility Patterns

Before illustrating our proposed mobility patterns, we first introduce the term *distribution distance*, which integrates heterogeneity and mobility. In FL, it is commonly assumed that the central server has knowledge of mobile clients' speeds and locations [11]. Additionally, FedDisco [6] assumes that the server is aware of each client's category-wise data distribution. Building on this, we assume that mobile clients in $\mathcal{C}_m$ have knowledge of the category-wise sample distributions and locations of static clients in $\mathcal{C}_s$. While these are strong assumptions, we will relax them in future work. These assumptions allow mobile clients to calculate the data distribution across different grid points in the network. Given the category-wise distribution of static clients and their locations, a mobile client $i_m \in \mathcal{C}_m$ can determine the data distribution at any location in $\mathcal{G}_{\mathcal{L}}$ before starting training. The category-wise distribution at a location $L \in \mathcal{G}_{\mathcal{L}}$, if the mobile client moves there, is given by:

$$D_{i_m,l}^{L} = \frac{\sum_{c\in\zeta_{i_m,L}} |\{(\boldsymbol{s}_n, y_n) \in \mathcal{D}_c \mid y_n = l\}|}{\sum_{c\in\zeta_{i_m,L}} |\mathcal{D}_c|}, l \in [Y], \tag{4}$$

where $\zeta_{i_m,L} := \{j \mid \|L_j^{(0)} - L\|_2 \le R_c, j \in \mathcal{C}_s\} \cup \{i_m\}$ denotes the set of fixed clients within the coverage area of the mobile client i_m at location L, including i_m itself, and $[Y]$ denotes the set of classification labels.

Using this category-wise distribution, we define the *distribution distance* from the client's current location $L_{i_m}^{(t-1)}$ (considered as the reference point) at the start of round t as:

$$d_{L_{i_m}^{(t-1)}}^{L} = \sqrt{\sum_{l\in[Y]} \left(D_{i_m,l}^{L} - D_{i_m,l}^{L_{i_m}^{(t-1)}}\right)^2}, \quad \forall L \in \mathcal{G}_{\mathcal{L}}. \tag{5}$$

1) Distribution-Aware Mobility (DAM): Given the distribution distances across all grid points with respect to the current location $L_{i_m}^{(t-1)}$ of mobile client i_m, our proposed **Distribution-Aware Mobility (DAM)** pattern assigns higher probabilities to locations with greater distribution distances. This encourages movement toward regions that mitigate the impact of data heterogeneity. Based on this probability distribution, the next desired location is determined. The probability of mobile client i_m moving to a location $L \in \mathcal{G}_{\mathcal{L}}$ at the end of local update of round t is given by:

$$p_{L|L_{i_m}^{(t-1)}}^{(t)} = \frac{d_{L_{i_m}^{(t-1)}}^{L}}{\sum_{L\in\mathcal{G}_{\mathcal{L}}} d_{L_{i_m}^{(t-1)}}^{L}}, \quad \forall L \in \mathcal{G}_{\mathcal{L}}. \tag{6}$$

Therefore, the desired new location of i_m can be sampled as: $L_{i_m}^{(des.,t)} \sim p_{L|L_{i_m}^{(t-1)}}^{(t)}$. However, the movement of mobile clients is restricted by a maximum allowable displacement per round R_m. If $\|L_{i_m}^{(des.,t)} - L_{i_m}^{(t-1)}\|_2 > R_m$, the client moves to the nearest location to $L_{i_m}^{(des.,t)}$ along its path. Formally, the updated location at the end of round t's local update is:

$$L_{i_m}^{(t)} = \begin{cases} L_{i_m}^{(des.,t)}; & \text{if } \|L_{i_m}^{(des.,t)} - L_{i_m}^{(t-1)}\|_2 \le R_m \\ \arg\min_{L\in\Gamma} \|L - L_{i_m}^{(des.,t)}\|_2; & \text{otherwise,} \end{cases} \tag{7}$$

where $\Gamma := \{L \in \mathcal{G}_{\mathcal{L}} \mid \|L - L_{i_m}^{(t-1)}\|_2 \le R_m\}$. Nevertheless, if the mobile client i_m does not reach its destination within the round, its destination remains unchanged until it reaches it. Once the desired location is reached, a new destination is drawn according the distribution in (6). Thus, the desired location can be expressed as:

$$L_{i_m}^{(des.,t)} \begin{cases} \sim p_{L|L_{i_m}^{(t-1)}}^{(t)}; & \text{if } L_{i_m}^{(t-1)} = L_{i_m}^{(des.,t-1)} \\ = L_{i_m}^{(des.,t-1)}; & \text{otherwise.} \end{cases} \tag{8}$$

2) Distribution-Aware Cluster-Center Mobility (DCM): The destination determined by the DAM mobility pattern—based on the probability distribution in Eq. (6)—can be any location within the grid, i.e., $L_{i_m}^{(des.,t)} \in \mathcal{G}_{\mathcal{L}}$. However, the large search space can potentially slow the convergence in mobility-assisted DFL. To address this, we propose

Distribution-Aware Cluster-Center Mobility (DCM) pattern, which reduces the destination search space and further enhances convergence. In DCM, mobile clients select destinations that correspond to the centers of clusters of static clients. These clusters are formed based on the communication radius R_c, and the cluster centers act as representative destinations. The selection process follows a structured approach:

- *Selecting the First Cluster Center*: The first cluster center L_{c_1} is sampled as a grid point $L_{c_1} \in \mathcal{G}_\mathcal{L}$ where a mobile client $i_m \in \mathcal{C}_s$ can communicate with the maximum number of neighboring static clients within radius R_c:
$$L_{c_1} \sim \arg\max_{L \in \mathcal{G}_\mathcal{L}} |\{j \in \mathcal{C}_s \mid \|L - L_j^{(0)}\|_2 \leq R_c\}|.$$
- *Selecting Subsequent Cluster Centers*: The second cluster center is chosen from the remaining static clients (i.e., excluding those already covered by the first cluster). If multiple locations have the same maximum number of neighbors, the one covering the most neighbors among all static clients is selected. In case of a tie, a location is randomly chosen. This process repeats until all static clients are assigned to a cluster.

Algorithm 1 formally describes this clustering procedure, ensuring an efficient selection of cluster centers $\mathcal{L}^c$.

Algorithm 1: Clustering

Inputs: Communication Radius R_c; Grid locations $\mathcal{G}_\mathcal{L}$; Static clients $\mathcal{C}_s$; Clients initial locations $L_i^{(0)} \in \mathcal{G}_\mathcal{L}, \forall i \in \mathcal{C}_s$.
Output: Cluster Center Locations $\mathcal{L}^c$.
$\mathcal{L}^c \leftarrow \{\}$; $\mathcal{C}_s^r \leftarrow \mathcal{C}_s$; $n = 1$
while $|\mathcal{C}_s^r| > 0$ **do**
  $\mathcal{L}_d^c = \arg\max_{L \in \mathcal{G}_\mathcal{L}} |\{j \in \mathcal{C}_s^r \mid \|L - L_j^{(0)}\|_2 \leq R_c\}|$
  if $|\mathcal{L}_d^c| > 1$ & $n > 1$ **then**
    $\mathcal{L}_d^c = \arg\max_{L \in \mathcal{L}_d^c} |\{j \in \mathcal{C}_s \mid \|L - L_j^{(0)}\|_2 \leq R_c\}|$
  $L_{c_n} \sim \mathcal{L}_d^c$
  $\mathcal{C}_s^r = \mathcal{C}_s^r \setminus \{j \in \mathcal{C}_s \mid \|L_{c_n} - L_j^{(0)}\|_2 \leq R_c\}$
  $\mathcal{L}^c = \mathcal{L}^c \cup \{L_{c_n}\}$; $n = n + 1$

In DCM, the mobile clients are destined to move across cluster centers $\mathcal{L}^c$, which reduces the number of possible desired locations compared to DAM. The distribution distance $d^{L^c}_{L_{i_m}^{(t-1)}}$ of each cluster center $L^c \in \mathcal{L}^c$ with respect to a mobile client i_m's current location $L_{i_m}^{(t-1)}$ at round t is computed using the same expression in Eq. (5). Thus, the probability distribution for movement across cluster centers, based on the distribution distance, is:

$$p^{(t)}_{L^c|L_{i_m}^{(t-1)}} = \frac{d^{L^c}_{L_{i_m}^{(t-1)}}}{\sum_{L^c \in \mathcal{L}^c} d^{L^c}_{L_{i_m}^{(t-1)}}}, \quad \forall L^c \in \mathcal{L}^c. \tag{9}$$

Due to the mobility constraint R_m, the movement of $i_m \in \mathcal{C}_m$ across $\mathcal{L}^c$ exploiting the probability distribution in Eq. (9) follows similar expressions as in Eq. (8) and Eq. (7) to obtain the desired and the updated locations at round t, respectively.

All the mobility patterns discussed are constrained by the movement constraint R_m. In the Random Movement pattern, mobile clients select random locations within the mobility constraint R_m. In contrast, with the proposed DAM and DCM patterns, mobile clients move toward their desired locations $L_{i_m}^{(des.,t)}$, and have to choose an intermediate location along the route if $\|L_{i_m}^{(des.,t)} - L_{i_m}^{(t-1)}\|_2 > R_m$ while performing local updates in round t. As a extreme case, these mobility patterns can be considered as **unconstrained movement** when $R_m = \infty$, allowing mobile clients to move freely to any desired location within the network in a given round.

V. Experimental Results

In this section, we conduct experimental evaluations to assess our proposed mobility patterns in a DFL system and compare their performance against **Static** and **Random Movement**, which serve as baselines.

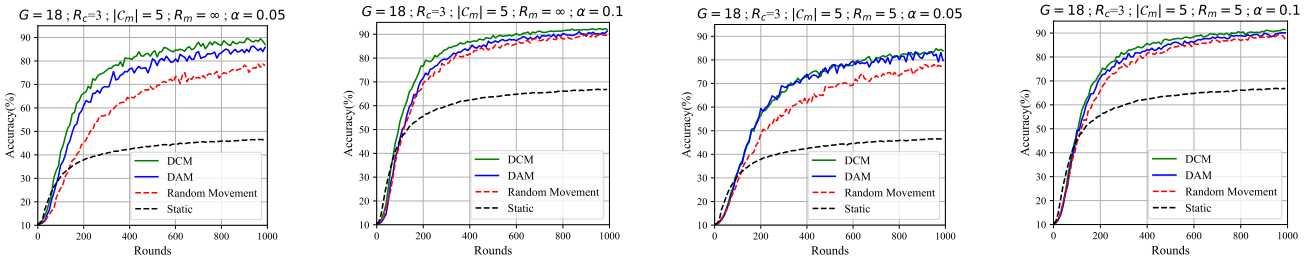
A. Simulation Setting

We divide our experiments into three different settings. In the first setting, we consider the MNIST classification task on a network of size 18×18 with 20 clients. In the second setting, we scale our simulations to a network of size 35×35 with 50 clients, again using the MNIST dataset, to evaluate the scalability of our framework. Finally, in the third setting, we assess the generalizability of our approaches by training on the CIFAR-10 dataset with 20 clients and a network of size 18×18. Since data heterogeneity is a crucial factor in DFL, we distribute the training samples among clients following a Dirichlet distribution with parameter α for local updates and use the corresponding test sets for evaluation. Unless otherwise specified, we consider communication radius $R_c = 3$ for the network topology.

For training on the MNIST dataset, we consider a simple model structure across the clients: two convolution layers followed by two fully connected layers. Training is performed using full-batch gradient descent with a learning rate of 0.3. For the CIFAR-10 dataset, in contrast, we adopt a ResNet50 model pretrained on ImageNet-1K, adapting the first convolutional layer (kernel size 3, stride 1) for CIFAR-10 input, and removing the initial MaxPool layer. Training on CIFAR-10 is conducted using mini-batch gradient descent with a batch size of 512 and a learning rate of 0.01. For both datasets, we use the SGD optimizer with momentum and weight decay values of 0.9 and 0.0005, respectively. To evaluate performance, we compute the average test accuracy across clients over 10 Monte Carlo runs, where models are trained on the respective training datasets (MNIST and CIFAR-10) and evaluated on the corresponding test sets every 10 global rounds.

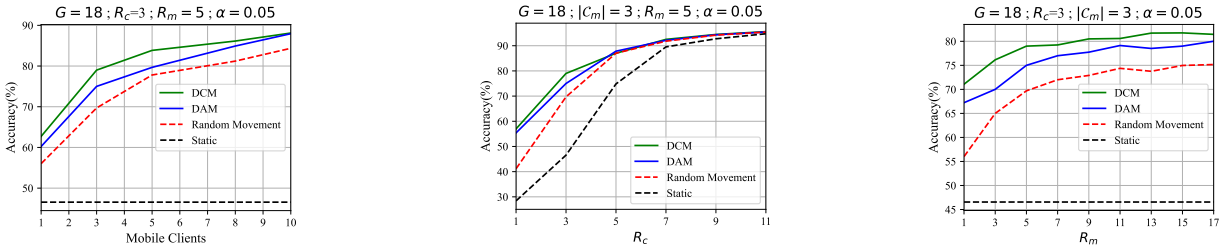
B. Results Analysis

Fig. 2 shows the DFL system's accuracy after 1,000 training rounds on MNIST dataset in a network of size 18×18, with 20 clients, including $|\mathcal{C}_m| = 5$ mobile clients. Fig. 2a and Fig. 2c show the results for $\alpha = 0.05$, while Fig. 2b and Fig. 2d correspond to $\alpha = 0.1$, representing a less heterogeneous scenario. Additionally, we compare the performance for both unconstrained ($R_m = \infty$) and constrained ($R_m = 5$) mobility.



(a) Unconstrained movements (b) Unconstrained movements (c) Constrained movements (d) Constrained movements

Fig. 2: Accuracy of DFL system on MNIST Dataset after 1000 rounds with 20 clients.



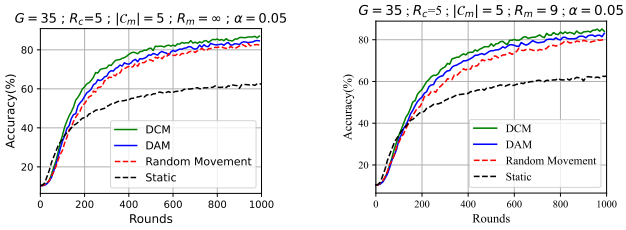
(a) Accuracy vs. $|\mathcal{C}_m|$ (b) Accuracy vs. R_c (c) Accuracy vs. R_m

Fig. 3: Impact of $|\mathcal{C}_m|$, R_c and R_m on accuracy of DFL system on MNIST Dataset after 1000 rounds with 20 clients.

The results show that introducing mobility into the network improves performance. Specifically, when a subset of clients are mobile and follow the Random Movement, the model achieves significantly higher accuracy compared to the Static scenario, where all users remain fixed at initial locations. Furthermore, considering Random Movement as a baseline, DAM and DCM improve accuracy on average by approximately 8% in both constrained and unconstrained movements, in highly heterogeneous settings ($\alpha = 0.05$). Similarly, when the data is distributed with less heterogeneity ($\alpha = 0.1$), DAM and DCM still achieve 3% accuracy improvement over the baseline. As expected, the proposed DCM outperforms DAM, and unconstrained mobility patterns lead to better convergence compared to their constrained counterparts.

1) Impact of Mobile Clients: To investigate further on the effect of the number of mobile clients $|\mathcal{C}_m|$ in the network, we consider the same learning task on the MNIST dataset in a network with 20 clients, data heterogeneity parameter $\alpha = 0.05$, and mobility constraint $R_m = 5$. As shown in Fig. 3a, increasing the number of mobile clients exposes more fixed clients to new updates, which enhances overall model performance. Additionally, both DCM and DAM consistently outperform Random Movement across different $|\mathcal{C}_m|$ values. As can be seen, DCM and DAM achieve, on average, 7% and 5.3% better accuracy, respectively, compared to Random Movement across diferent values of $|\mathcal{C}_m|$.

2) Impact of Communication Radius (R_c): Fig. 3b illustrates the impact of communication range in a network with 20 clients, where $|\mathcal{C}_m| = 3$ mobile clients are deployed in a network, and data heterogenity parameter $\alpha = 0.05$, and the mobility constraint is set to $R_m = 5$. As expected, with a larger communication range, more clients are allowed to exchange their model updates. Therefore, higher accuracy is obtained. However, it can be seen that if the mobile clients are utilized with our proposed mobility patterns, they can outperform Random Movement significantly in networks with low connectivity. For example, at $R_c = 1$, DCM and DAM achieve around 15% higher accuracy than Random Movement.

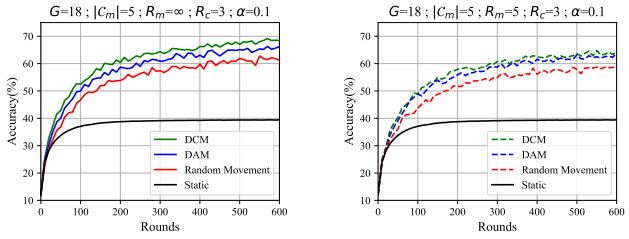


(a) Unconstrained movements (b) Constrained movements

Fig. 4: Accuracy of DFL with 50 clients after 1000 rounds in a larger network with 5 mobile clients using MNIST dataset.

3) Impact of Mobility Constraint (R_m): To investigate the effect of client mobility constraint R_m, we conduct the simulation in a network with 20 clients and 3 mobile clients with $\alpha = 0.05$ for data heterogeneity and repeat these experiments while utilizing the mobile clients at different speeds. Fig. 3c illustrates the impact of R_m and shows that accuracy improves with the increase of R_m and gradually approaches the performance of mobile clients operating without movement constraints in the same setting. This is because mobile clients with higher R_m are allowed to explore the network with more freedom and provide the fixed clients with more diverse and useful updates.

4) Larger Network: To evaluate the scalability of our proposed mobility patterns, we simulate the DFL system in a larger network with a size of 35×35 and 50 clients by utilizing 5 mobile clients and mobility constraint $R_m = 9$. We set $R_c = 5$ and considered $\alpha = 0.05$ to investigate our proposed



(a) Unconstrained movements (b) Constrained movements

Fig. 5: Accuracy of DFL with 20 clients after 600 rounds in a network with 5 mobile clients using CIFAR-10 dataset.

framework in a highly heterogeneous situation. Fig. 4 illustrates the accuracy of DFL in 1,000 rounds with this setting. The proposed DAM and DCM mobility patterns outperform Random Movement in both unconstrained and constrained settings. Specifically, DCM achieves approximately 5% and 4% higher accuracy than Random Movement in the uncostrained and constrained scenarios, respectively.

5) Performance on CIFAR-10: We conduct additional experiments on the CIFAR-10 dataset using a 18×18 network with 20 clients, $R_c = 3$, and $\alpha = 0.1$. We compare the performance of the proposed DAM and DCM under both constrained ($R_m = 5$) and unconstrained movement scenarios. As in Fig. 5, mobility significantly enhances convergence for CIFAR-10, mirroring the improvements observed on MNIST. Furthermore, in the unconstrained scenario, DAM and DCM achieve approximately 5% and 8% higher accuracy than random movement, while in the constrained setting, they offer around 4% and 5% higher accuracy, respectively.

VI. Conclusion and Future Works

In this work, we explored the impact of mobility in DFL systems and demonstrated that the presence of mobile clients, even with random movement, significantly improves accuracy. To this end, we proposed two mobility patterns, DAM and DCM, which allow the mobile client to explore the network more effectively, leveraging the data distribution and location of static clients, than the baselines. Experimental results demonstrate that in highly heterogeneous data settings, both DAM and DCM outperform random movement, with DCM achieving particularly notable accuracy improvements. For future work, we plan to theoretically analyze convergence and relax the assumption that mobile clients know static clients' data distributions and locations, exploring mobility's impact.

References

[1] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of things journal*, vol. 3, no. 6, pp. 854–864, 2016.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[3] C. Feng, H. H. Yang, D. Hu, T. Q. Quek, Z. Zhao, and G. Min, "Federated learning with user mobility in hierarchical wireless networks," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 01–06.

[4] Y. Wang, H. Fu, R. Kanagavelu, Q. Wei, Y. Liu, and R. S. Mong Goh, "An aggregation-free federated learning for tackling data heterogeneity," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 26 223–26 232.

[5] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[6] R. Ye, M. Xu, J. Wang, C. Xu, S. Chen, and Y. Wang, "Feddisco: Federated learning with discrepancy-aware collaboration," in *International Conference on Machine Learning*. PMLR, 2023, pp. 39 879–39 902.

[7] L. Barbieri, S. Savazzi, M. Brambilla, and M. Nicoli, "Decentralized federated learning for extended sensing in 6g connected vehicles," *Vehicular Communications*, vol. 33, p. 100396, 2022.

[8] E. T. M. Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2983–3013, 2023.

[9] T. Zhu, F. He, L. Zhang, Z. Niu, M. Song, and D. Tao, "Topology-aware generalization of decentralized sgd," in *International Conference on Machine Learning*. PMLR, 2022, pp. 27 479–27 503.

[10] J. Wang, A. K. Sahu, Z. Yang, G. Joshi, and S. Kar, "Matcha: Speeding up decentralized sgd via matching decomposition sampling," in *2019 Sixth Indian Control Conference (ICC)*. IEEE, 2019, pp. 299–300.

[11] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, and M. S. Hossain, "Mobility-aware proactive edge caching for connected vehicles using federated learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5341–5351, 2020.

[12] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized sgd with changing topology and local updates," in *International conference on machine learning*. PMLR, 2020, pp. 5381–5393.

[13] Y. Jiang, H. Kang, J. Liu, and D. Xu, "On the convergence of decentralized stochastic gradient descent with biased gradients," *IEEE Transactions on Signal Processing*, vol. 73, pp. 549–558, 2025.

[14] Y. Chen, K. Yuan, Y. Zhang, P. Pan, Y. Xu, and W. Yin, "Accelerating gossip sgd with periodic global averaging," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1791–1802.

[15] C. Li, G. Li, and P. K. Varshney, "Decentralized federated learning via mutual knowledge transfer," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1136–1147, 2021.

[16] A. Bellet, A.-M. Kermarrec, and E. Lavoie, "D-cliques: Compensating for data heterogeneity with topology in decentralized federated learning," in *2022 41st International Symposium on Reliable Distributed Systems (SRDS)*. IEEE, 2022, pp. 1–11.

[17] Y. Dandi, A. Koloskova, M. Jaggi, and S. U. Stich, "Data-heterogeneity-aware mixing for decentralized learning," *arXiv preprint arXiv:2204.06477*, 2022.

[18] B. Le Bars, A. Bellet, M. Tommasi, E. Lavoie, and A.-M. Kermarrec, "Refined convergence and topology learning for decentralized sgd with heterogeneous data," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 1672–1702.

[19] M. F. Pervej, R. Jin, and H. Dai, "Resource constrained vehicular edge federated learning with highly mobile connected vehicles," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 6, pp. 1825–1844, 2023.

[20] C. Feng, H. H. Yang, D. Hu, Z. Zhao, T. Q. Quek, and G. Min, "Mobility-aware cluster federated learning in hierarchical wireless networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 10, pp. 8441–8458, 2022.

[21] Y. Peng, X. Tang, Y. Zhou, Y. Hou, J. Li, Y. Qi, L. Liu, and H. Lin, "How to tame mobility in federated learning over mobile networks?" *IEEE Transactions on Wireless Communications*, vol. 22, no. 12, pp. 9640–9657, 2023.

[22] J. Bian and J. Xu, "Accelerating asynchronous federated learning convergence via opportunistic mobile relaying," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 7, pp. 10 668–10 680, 2024.

[23] Y. Qu, H. Dai, Y. Zhuang, J. Chen, C. Dong, F. Wu, and S. Guo, "Decentralized federated learning for uav networks: Architecture, challenges, and opportunities," *IEEE Network*, vol. 35, no. 6, pp. 156–162, 2021.

[24] S. de Heredia, "Exploring the impact of client mobility on decentralized federated learning systems," Ph.D. dissertation, Delft University of Technology, 2024.