SGFusion: Stochastic Geographic Gradient Fusion in Federated Learning

Khoa Nguyen[§*], Khang Tran[§*], NhatHai Phan[§], Cristian Borcea[§]
Ruoming Jin[¶], Issa Khalil[ℑ]
[§]New Jersey Institute of Technology, Newark, NJ, USA, [¶]Kent State University, Kent, OH, USA
[ℑ]Qatar Computing Research Institute, HBKU, Doha, Qatar
E-mail: {nk569, kt36, phan, borcea}@njit.edu, rjin1@kent.edu, ikhalil@hbku.edu.qa
[*]Co-first authors.

***Abstract*—This paper proposes Stochastic Geographic Gradient Fusion (SGFusion), a novel training algorithm to leverage the geographic information of mobile users in Federated Learning (FL). SGFusion maps the data collected by mobile devices onto geographical zones and trains one FL model per zone, which adapts well to the data and behaviors of users in that zone. SGFusion models the local data-based correlation among geographical zones as a hierarchical random graph (HRG) optimized by Markov Chain Monte Carlo sampling. At each training step, every zone fuses its local gradient with gradients derived from a small set of other zones sampled from the HRG. This approach enables knowledge fusion and sharing among geographical zones in a probabilistic and stochastic gradient fusion process with self-attention weights, such that *"more similar"* zones have *"higher probabilities"* of sharing gradients with *"larger attention weights."* SGFusion remarkably improves model utility without introducing undue computational cost. Extensive theoretical and empirical results using a heart-rate prediction dataset collected across 6 countries show that models trained with SGFusion converge with upper-bounded expected errors and significantly improve utility in all countries compared to existing approaches without notable cost in system scalability.**

***Index Terms*—Geographical FL, Differential Privacy**

I. Introduction

Although Federated learning (FL) [1] has many applications for mobile users [2], we still need to find a practical solution that obtains good model accuracy while adapting to user mobility behavior, scales well as the number of users increases, and protects user data privacy, which is especially important for mobile sensing applications (e.g., mobile health). A common approach toward this goal is to group users into different clusters, each of which is trained in an FL manner to achieve better model performance [3]–[5]. There are different clustering criteria, such as using the objective function on the users' local data distribution [3], [6], the gradient similarity [4], [5] or the users' geographical location [7] to reduce the discrepancy of model utility among users and clusters. Among these approaches, leveraging users' geographical information is the most suitable approach to divide the physical space into geographical zones of users mapped to a mobile-edge-cloud FL architecture [7] since it scales well with the increasing number of users in real-world settings. By enabling geographical zones of (mobile) users to share gradients with their geographical neighboring (adjacent) zones, FL has shown outstanding performance in model utility and server scalability in real-world deployments of mobile sensing applications, e.g., heart rate prediction and human activity recognition.

Challenges. Although leveraging users' geographical information is promising for real-life FL deployments on mobile devices, the trade-off between model utility and system scalability has yet to be addressed in a systematic way. Specifically, there is a lack of optimized geographical training algorithms. As a result, geographical zones without sufficient training data or appropriate geographical correlations with other zones often have poor model utility.

Addressing this problem is challenging, given the complex and dynamic correlation among geographical zones. In fact, a specific zone can obtain shared gradients from all other zones. However, it will significantly increase computational complexity on all users and edge devices managing geographical zones while offering negligible model utility improvements. The trade-off between model utility and system scalability raises a fundamental question: *"How to share gradients among geographical zones to optimize model utility without affecting system scalability?"*

Contributions. To systematically answer this question, we propose Stochastic Geographic Gradient Fusion (**SGFusion**), a novel FL training algorithm for mobile users. SGFusion models the local data-based correlation among geographical zones of users as a hierarchical random graph (HRG) [8], optimized by Markov Chain Monte Carlo sampling. HRG represents the probability for one zone to share its gradient with another zone. At a training step, every zone can fuse its local gradients with gradients derived from a set of other zones sampled from the HRG. This approach enables knowledge fusion and sharing among geographical zones in a probabilistic and stochastic gradient fusion process with dynamic attention weights, such that *"more similar"* zones have *"higher probabilities"* of sharing gradients with *"larger attention weights."* In fact, using HRG reduces and structures the search space, allowing us to identify similar zones better. Zone sampling enables us to reduce the computational cost further. As a result, SGFusion can remarkably improve model utility without introducing undue computational cost.

Extensive theoretical and empirical results show that models trained with SGFusion converge with upper-bounded expected errors. SGFusion significantly improves model utility without notable cost in system scalability compared with existing

approaches. The experiments on heart rate prediction demonstrate that, among the total of 115 zones across 6 countries, **more than double the number of zones benefit from SGFusion** compared with the state-of-the-art clustering-based FL approaches and their variants [7], [9], without slowing the convergence process. SGFusion improves the aggregated model utility across six countries by 3.23% compared with existing approaches. Here is the code for SGFusion: **Code**.

II. Background and Related Work

In FL, a coordination server and a set of N users jointly train a model h_θ, where θ is a vector of model weights [1].

Clustering-based FL. Instead of training one global model [1], the service provider in clustering-based FL divides the users into clusters and trains an FL model for each cluster to enhance the model's performance under non-independent and identically distributed (non-IID) data distribution across users [10]–[14]. The challenges of non-IID data could also be mitigated through new aggregating methods [15]–[17]. A pioneering work in this setting [10] leverages the hierarchical clustering method to cluster the clients based on their updating gradients. Similarly, Ghosh et al. [11] proposed sending multiple models associated with different data distribution to the clients and letting them choose the model that minimizes their data loss. Although these works mitigate the non-IID problems, they incur high computation overhead on the users' devices proportional to the number of clusters, hence limiting the scalability of the existing systems.

Decentralized FL. Another line of work addressing the scalability problem proposes to replace the centralized coordinating server with multiple servers, each of which being associated with a cluster [5], [18]–[21]. For instance, [5] proposed a multi-center FL setting to personalize FL models to different users' data distribution while avoiding high computation overhead. Similarly, Zhang et al. [19] introduced a three-layer collaborative FL architecture with multiple edge servers to learn ML models for IoT devices in FL settings. However, existing decentralized FL approaches usually disregard the users' geographical locations; as a result, they increase communication overhead and do not adapt well to user mobility behaviors in a mobile-edge-cloud FL architecture in practice.

Zone FL is a recent effort to make FL deployments practical in real-world scenarios, which aims at high model utility by adapting well to user mobility behavior while scaling well to the increasing number of users [7]. To achieve this goal, Zone FL divides the physical space into non-overlapping geographical zones and maps the correlation between users, zones, and FL models to a mobile-edge-cloud FL architecture.

In this setting, an edge-device manages a zone-FL model trained for users whose local data is collected in that zone on their mobile devices. The cloud manages the general zone structure, such as zone granularity. Given a specific zone z, the goal is to train an FL model for the zone, θ_z, minimizing an objective function $F_z = \frac{1}{m_z}\sum_{u\in z}\ell(\theta_z, D_u)$, where $\ell(\cdot,\cdot)$ is a loss function (e.g., cross-entropy function) and m_z is the number of users in z, using data D_u collected by users u in the zone z: $\theta_z^* = \arg\min_{\theta_z} F_z$. Given a set of zones $z \in Z$ where $|Z|$ denotes the number of zones, the goal of Zone FL is to find a set of $\{\theta_z^*\}_{z\in Z}$ that minimizes the average objective function of all the zones, as follows:

$$\{\theta_z^*\}_{z\in Z} = \arg\min_{\{\theta_z\}_{z\in Z}} \frac{1}{|Z|}\sum_{z\in Z} F_z. \tag{1}$$

To enhance model utility by fostering knowledge fusion among zone models in Eq. 1, the current training algorithm in Zone FL, called Zone Gradient Diffusion (ZGD), enables every zone to fuse its local gradient with gradients derived from its geographical neighboring/adjacent zones. At round t, the zone z sends its model weights θ_z^t to its neighboring zones, denoted as $\mathcal{N}(z)$. Then, the neighboring zones derive their local gradients by using the shared model weight θ_z^t and their local data, as follows:

$$\forall z' \in \mathcal{N}(z) : \nabla_{\theta_z^t} F_{z'} = \frac{1}{m_{z'}}\sum_{u\in z'} \nabla_{\theta_z^t}\ell(\theta_z^t, D_u), \tag{2}$$

where $\nabla_{\theta_z^t}\ell(\theta_z^t, D_u)$ is the gradient derived by the user u using the model weight θ_z^t and their local data D_u.

After receiving all the gradients $\{\nabla_{\theta_z^t} F_{z'}\}$ from neighboring zones $z' \in \mathcal{N}(z)$, the zone z will fuse these gradients associated self-attention coefficients $\{\lambda_{z,z'}\}$ with its local gradient to update its zone model, as follows:

$$\theta_z^{t+1} = \theta_z^t - \eta_t\big[\nabla_{\theta_z^t} F_z + \sum_{z'\in\mathcal{N}_z^t} \lambda_{z,z'}\nabla_{\theta_z^t} F_{z'}\big], \tag{3}$$

where η_t is the learning rate at round t, and the self-attention coefficients capture the normalized similarities between the local gradient of z and the shared gradients from its neighboring zones, as follows: $\lambda_{z,z'} = \frac{\exp(e_{z,z'})}{\sum_{\tilde{z}\in\mathcal{N}(z)}\exp(e_{z,\tilde{z}})}$, where $e_{z,z'} = \sigma\big(\langle\nabla_{\theta_z^t} F_z; \nabla_{\theta_z^t} F_{z'}\rangle\big)$, $\sigma(\cdot)$ is the sigmoid function and $\langle\cdot;\cdot\rangle$ is the inner product.

The key idea of Eq. 3 is that the *"more similar"* the gradients of a neighboring zone z' are with those of zone z, the *"higher the coefficient"* $\lambda_{z,z'}$ is; thus, resulting in a larger influence of zone z' on the model training of zone z.

III. Stochastic Geographic Gradient Fusion

Although Zone FL is better at addressing the trade-off between model utility and system scalability compared with classical FL, there is still a fundamental question that it did not address: *"With which zones should a zone z fuse its local gradients at a given training round to achieve high model utility without undue computational cost, and how?"* Answering this question is non-trivial. First, fusing knowledge via shared gradients from neighboring zones does not necessarily improve the model utility of a specific zone, given diverse local data distributions among these zones. It is well-known that diverse local data distributions can cause scattered gradients, thus degenerating the FL model utility [22]. Second, a deterministic gradient descent (GD) fusion approach, which uses fixed neighboring zones across training rounds, is not well-optimized since it does not consider the

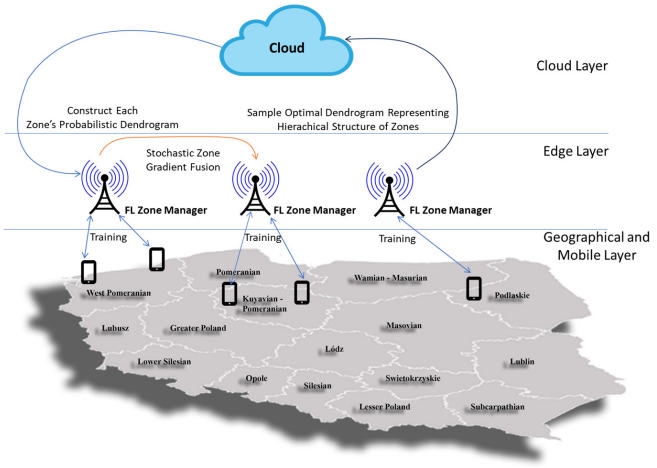


Fig. 1: SGFusion with geographical zones.

correlation between zone z and all the other zones. A naive solution to this problem is to fuse a zone z's local gradients with the gradients from all geographical zones at each training round. However, applying this (deterministic) GD significantly increases the training cost of $|Z|$ zone-FL models after T training rounds by $|Z|^2 \times N$ times compared with classical FL training [1], where N is the total number of users in all the zones, and $\frac{|Z|^2 \times N}{\sum_{z \in Z} \sum_{z' \in \mathcal{N}(z)} m'_z}$ times compared with ZGD [7]. Therefore, existing training algorithms either degrade the model utility or affect the system scalability.

SGFusion Overview. To address these problems, we propose stochastic geographic gradient fusion (SGFusion), a novel FL training algorithm for mobile users. SGFusion uses a hierarchical random graph (HRG) [8] to model the correlations among zones as sampling probabilities based on the distances between their local data distributions. HRG is scalable due to its ability to efficiently represent statistical correlations among zones when the number of zones increases, making it efficient for settings with large numbers of users. Then, SGFusion optimizes the HRG by Markov Chain Monte Carlo (MCMC) sampling [23]. At each training round, each zone z samples a small set of zones given the HRG to fuse its local gradient with local gradients derived from these zones. This method enables knowledge fusion and sharing among zones, such that *"more similar"* zones have *"higher probabilities"* of sharing gradients with *"larger attention weights"* at each training round of each zone-FL model. As a result, SGFusion reduces the data diversity and scattered shared gradients while providing sufficient knowledge fused from other zones to improve zone models $\{\theta_z^*\}_{z \in Z}$ in Eq. 1.

A. Geographic HRG & Probabilistic Dendrograms

Figure 1 illustrates the general framework of SGFusion. Given a set of users $u \in z$, each user u collects data $D_u = \{(x, y)\}$ in the zone z, where x and y are the input and label of a data sample (x, y), respectively. To build the HRG, each user

Algorithm 1 Geographic HRG

Input: Fully connected graph G
Output: Probabilistic dendrograms

1: **Dendrogram Sampling and Optimizing Process:**
2: Randomly initialize the dendrogram $\mathcal{T}$
3: **while** not convergence of $\mathcal{T}$ **do**
4: **Randomly select** an internal node $r \in \mathcal{T}$
5: **Uniformly select** either α or β-transition given r to create the new dendrogram candidate $\mathcal{T}'$
6: **Sample** the transition $\mathcal{T} \rightarrow \mathcal{T}'$ using Eq. 7.
7: **Construct the Probabilistic Dendograms:**
8: **Retrieve** a set of internal ancestor nodes of z, denoted as S_z, given $\mathcal{T}$
9: **Initialize** $\mathcal{T}_z$ given $\mathcal{T}$
10: **Compute** sampling probabilities $\forall r \in S_z : p_r = \frac{\exp(-d_r)}{\sum_{r \in S_z} \exp(-d_r)}$
11: **Update** the probability value of internal node r in $\mathcal{T}_z$ with p_r
12: **Return:** $\mathcal{T}_z$

u independently sends their data label distribution[1], denoted as $\mathcal{Y}_u$, aggregated from all labels $\{y\}$ to their corresponding edge-device managing the zone z. The edge device of zone z averages these local data label distributions to create a zone-level data label distribution $\mathcal{Y}_z$, as follows:

$$\forall z \in Z : \mathcal{Y}_z = \frac{1}{m_z} \sum_{u \in z} \mathcal{Y}_u. \tag{4}$$

All the edge devices send their zone-level data label distributions $\{\mathcal{Y}_z\}_{z \in Z}$ to the cloud independently. Now, the cloud can construct a fully connected graph G in a centralized manner, in which a node represents a zone and an edge represents the distance, e.g., Euclidean distance [25], between two zones z and z' using their zone-level data label distributions $d(\mathcal{Y}_z, \mathcal{Y}_{z'})$. Other distance functions such as Manhattan distance [26] and Minkowski distance [27] could be used to calculate the distance between two zones z and z'.

Given the graph G, Alg. 1 (Lines 1-6) describes how to construct the Geographic HRG and the dendrograms. The cloud randomly samples a hierarchical structure of the zones as a tree dendrogram $\mathcal{T}(\{r, d_r\})$, consisting of $|Z|$ zones as leaf nodes and a set of internal nodes r associated with average distance scores d_r between their left and right sub-trees, L_r and R_r, as follows:

$$\forall r \in \mathcal{T} : d_r = \frac{\sum_{z \in L_r, z' \in R_r} d(\mathcal{Y}_z, \mathcal{Y}_{z'})}{n_{L_r} n_{R_r}}, \tag{5}$$

where n_{L_r} and n_{R_r} are the numbers of zones in the left and the right sub-trees of the internal node r.

[1]The distribution can have different forms, such as histograms or class distributions, depending on the downstream learning tasks. Sending a data label histogram to the edge-device can incur a privacy risk, which can be effectively addressed by using the Laplace mechanism to preserve differential privacy [24], as in the III-B, without affecting model utility notably.

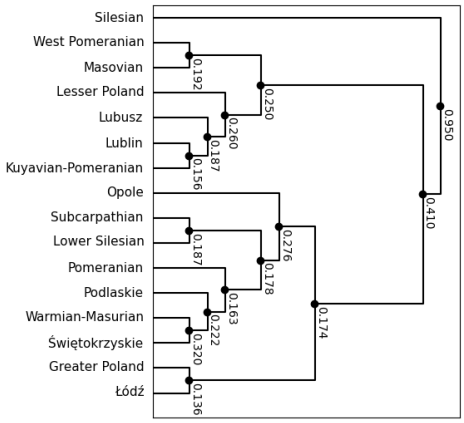


Fig. 2: Dendrogram $\mathcal{T}$ with 16 zones in Poland (as shown in Figure 1) using Euclidean distance.

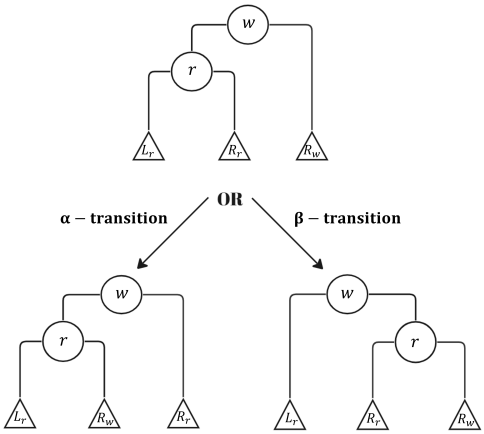


Fig. 3: Given a current state of an internal node r, there are only two possible candidate states, which are the result of α-transition and β-transition on the node r.

Figure 2 illustrates an example of a dendrogram $\mathcal{T}$ with 16 zones as leaf and internal nodes, each of which is associated with a value quantifying the average Euclidean distance between zones from its left and right sub-trees. For example, in the Heart Rate Prediction dataset (HRP) [28] that we use in our experiments, $\mathcal{Y}_z$ is the average histogram distribution of the heart rate of all users in the zone z.

To optimize the dendrogram $\mathcal{T}$, SGFusion applies the MCMC sampling to minimize the total average distances in all the internal nodes, indicating that $\mathcal{T}$ can be used to reconstruct the original graph G with minimized loss. In other words, the dendrogram $\mathcal{T}$ is optimized to represent the correlations among zones based on their local data label distributions. The optimization objective of $\mathcal{T}$ is as follows:

$$\mathcal{T}^* = \arg\min_{\mathcal{T}} \mathcal{L}(\mathcal{T}), \tag{6}$$

where the utility loss of $\mathcal{T}$ is computed by $\mathcal{L}(\mathcal{T}) = \sum_{r \in \mathcal{T}} d_r$.

At each state, the MCMC sampling picks a random internal node r to re-arrange the structure of its three associated sub-trees, consisting of its children subtrees, i.e., L_r and R_r, and a sibling subtree, i.e., R_w (Figure 3). In the α-transition, given an internal node r, the order of the three subtrees associated with r is rearranged while keeping the state of r. On the other hand, in the β-transition, the order of the three subtrees does not change, but the state of r is changed compared with w. SGFusion uniformly chooses either α-transition or β-transition with an equal probability of 0.5 to create a new dendrogram candidate $\mathcal{T}'$. Then, SGFusion updates $\mathcal{T}$ with a probability $\rho = \min(1, \frac{\exp(\mathcal{L}(\mathcal{T}))}{\exp(\mathcal{L}(\mathcal{T}'))})$, as follows:

$$\mathcal{T} = \begin{cases} \mathcal{T}', & \text{with probability } \rho \\ \mathcal{T}, & \text{with probability } 1-\rho. \end{cases} \tag{7}$$

The key idea of updating the dendrogram using Eq. 7 is that SGFusion always accepts the transition from $\mathcal{T}$ to $\mathcal{T}'$ when it minimizes the utility loss $\mathcal{L}(\mathcal{T})$; otherwise, SGFusion accepts the transition that increases the utility loss $\mathcal{L}(\mathcal{T})$ with the probability of $\frac{\exp(\mathcal{L}(\mathcal{T}))}{\exp(\mathcal{L}(\mathcal{T}'))}$. SGFusion executes the MCMC sampling process until the convergence of $\mathcal{T}$. By doing so, SGFusion optimizes the dendrogram to present the hierarchical structure of all the geographical zones.

Algorithm 2 SGFusion Training Algorithm

Input: Zone z and Probabilistic Dendrogram $\mathcal{T}_z$
Output: Zone-FL model θ_z

1: **Initialize** the zone's model weight θ_z^0
2: **for** $t = 1, 2, \ldots, T$ **do**
3: **Sample** a set of zones $\mathcal{N}(z,t)$ given $\mathcal{T}_z$
4: $\forall z' \in \mathcal{N}(z,t) : e_{z,z'} \leftarrow \sigma\big(\nabla_{\theta_z^t} F_z; \nabla_{\theta_z^t} F'_z\big)$
5: $\forall z' \in \mathcal{N}(z,t) : \lambda_{z,z'} \leftarrow \frac{\exp(e_{z,z'})}{\sum_{\tilde{z} \in \mathcal{N}(z,t)} \exp(e_{z,\tilde{z}})}$
6: $\theta_z^{t+1} \leftarrow \theta_z^t - \eta_t \big[\nabla_{\theta_z^t} F_z + \sum_{z' \in \mathcal{N}(z,t)} \lambda_{z,z'} \nabla_{\theta_z^t} F_{z'}\big]$
7: **Return:** θ_z^T

Probabilistic Dendrogram (Alg.1 Lines 7-12). After building the dendrogram $\mathcal{T}$, the cloud starts constructing a probabilistic dendrogram, denoted as $\mathcal{T}_z$, for each zone z, in which the value of an internal node r is the probability p_r of zones in either the left sub-tree or the right sub-tree of r to share gradients with the zone z at a training round. To construct the probabilistic dendrogram $\mathcal{T}_z$ for a zone z, the cloud creates $\mathcal{T}_z$ as a copy of the dendrogram $\mathcal{T}$ with empty values for internal nodes. Then, the cloud computes the probabilities $\{p_r\}$ for ancestors of zone z, denoted as S_z, by normalizing distance scores d_r of these ancestors:

$$\forall r \in S_z : p_r = \exp(-d_r) / \big(\sum_{r \in S_z} \exp(-d_r) \big), \tag{8}$$

where p_r is the probability value of an internal node r in $\mathcal{T}_z$ and $\forall z : \sum_{r \in S_z} p_r = 1$. For instance, Figure 4 shows the probabilities of zones sharing their local gradients with the zone "West Pomeranian" at a training round. Based on the heart rate dataset, zone "Lublin" has a probability of 0.293 to share its local gradient with "West Pomeranian."

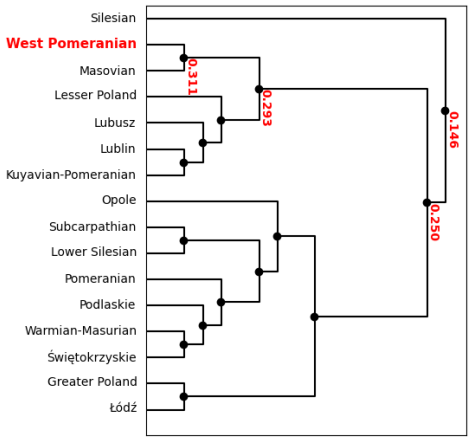


Fig. 4: Probabilistic dendrogram of zone "West Pomeranian" in Poland derived from Figure 2.

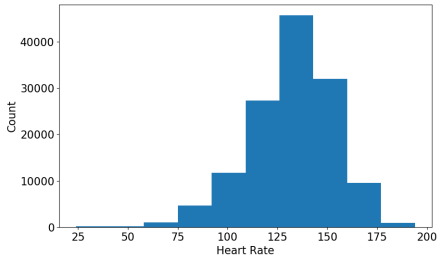


Fig. 5: Data label distribution $\mathcal{Y}_u$ of a particular user u.

B. DP-preserving Local Data Label Histogram

Differential privacy (DP) [24] can be used to release the statistical information of the datasets while protecting individual data privacy. Given a randomized algorithm $\mathcal{M}$, if for any two adjacent input datasets D and D' which differ by only one data point, and for all possible outputs $\mathcal{O} \in Range(\mathcal{M})$, where $Range(\mathcal{M})$ denotes every possible output of $\mathcal{M}$:

$$Pr[\mathcal{M}(D) = \mathcal{O}] \leq e^{\epsilon} Pr[\mathcal{M}(D') = \mathcal{O}], \tag{9}$$

then, $\mathcal{M}$ satisfies the ϵ-DP.

The difference in the distribution between D and D' is controlled by the privacy budget ϵ. A smaller ϵ enforces a stronger privacy guarantee. Due to the risk that labeled data points could leak the user's raw data to the edge devices, SGFusion adapts this mechanism to protect the data label distribution of a user (Figure 5). Given a data label distribution $\mathcal{Y}_u$ of a user u, the global sensitivity Δh_u of the data label histogram is defined as follows:

$$\Delta h_u = \max_{y} |\mathcal{Y}_u - \mathcal{Y'}_u|, \tag{10}$$

where $\mathcal{Y}_u$ and $\mathcal{Y'}_u$ and the data label histograms derived from theQ data D and D', which are different by at most one data point's label y. Then, we add the Laplace noise $Lap(0, \frac{\Delta h_u}{\epsilon})$ into $\mathcal{Y}_u$ to preserve the DP of the user's data label histogram:

$$\mathcal{M}(\mathcal{Y}_u, \epsilon) = \mathcal{Y}_u + Lap(0, \frac{\Delta h_u}{\epsilon}). \tag{11}$$

Doing so guarantees this mechanism satisfies ϵ-DP following [24]. We apply this mechanism to independently derive every user's DP-preserving data label histogram. We then use the perturbed histograms to construct the HRG and the probabilistic dendrograms (as in the SGFusion algorithm without privacy protection). Every user u independently sends their DP-preserving data label distribution $\mathcal{M}(\mathcal{Y}_u, \epsilon)$ to their corresponding edge-device managing the zone z, and this device aggregates these data label histograms to create a zone-level data label distribution $\mathcal{Y}_z$, as follows:

$$\forall z \in Z : \mathcal{Y}_z = \frac{1}{m_z} \sum_{u \in z} \mathcal{M}(\mathcal{Y}_u, \epsilon). \tag{12}$$

Then, all the edge devices send their data label distribution to the cloud independently to construct a fully connected graph G, in which a node represents a zone and an edge represents the distance $d(\mathcal{Y}_z, \mathcal{Y}_{z'})$ (e.g., Euclidean distance, Manhattan distance, or Minkowski distance) between two zones z and z'. Next, with the graph G, the cloud constructs the Geographic HRG and the dendrogram based on Alg. 1 (Lines 1-6). After building the dendrogram, the cloud starts constructing the probabilistic dendrograms (Alg.1 Lines 7-12).

We found that there is a marginal cost to protect the label data. For example, in our experiments, given $\epsilon = 10$, preserving the data label histograms of all users causes 0.15% difference in Norway's model utility, from 20.04 to 20.07 in the RMSE test loss, while outperforming the other baselines. This cost is low because the aggregated histogram neutralizes the noise added to the users' data label histograms. Therefore, SGFusion still has a good model utility while deriving DP-preserving data label histograms.

C. SGFusion Training

After constructing all probabilistic dendrograms $\{\mathcal{T}_z\}_{z \in Z}$, the cloud sends them to the corresponding edge servers for training zone models. For brevity, let us describe SGFusion training for a zone z, given its associated dendrogram $\mathcal{T}_z$, since the training is done independently for each zone.

At each round t, we propose a bottom-up sampling algorithm, as in Figure 6 and Alg. 2, for zone z (a leaf node of the probabilistic dendrogram $\mathcal{T}_z$) to sample a set of zones used for its gradient fusion. From the leaf node represented by zone z, our algorithm will travel to every (internal) ancestor node r of z, from the closest ancestor node to the root node, and sample zones in r's sub-trees with the probability p_r associated with the node r. Each zone is only sampled once in this process. As a result, zone z identifies a set of zones $\mathcal{N}(z,t)$ at training round t to fuse its local gradients with these sampled zones' local gradients in a self-attention mechanism and update the zone model θ_z, as follows:

$$\theta_z^{t+1} \leftarrow \theta_z^t - \eta_t \big[\nabla_{\theta_z^t} F_z + \sum_{z' \in \mathcal{N}(z,t)} \lambda_{z,z'} \nabla_{\theta_z^t} F_{z'}\big]. \tag{13}$$

SGFusion trains all zone-FL models $\{\theta_z\}$ using Eq. 13 independently until the models converge after T training rounds.

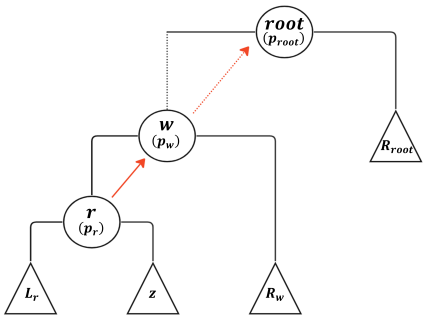


Fig. 6: Bottom-up zone sampling algorithm.

IV. Theoretical Guarantees

A. Convergence Analysis

In this section, we analyze the convergence rate of SGFusion. First, it is worth noting that the HRG is only sampled once before the training process of SGFusion, which is a preprocessing step. Since SGFusion computes the distance between two zones based on Euclidean metric, the complete graph across the zones is an undirected graph, which ensures the HRG sampling process is reversible and ergodic (i.e., any pairs of Dendograms can be transformed to each other with finite sampling steps). Therefore, the HRG sampling process has a unique stationary distribution after it converges to equilibrium [8], which is reached in our experiments.

Given a converged HRG representing the relation across the zones, we analyze SGFusion's convergence rate when optimizing a strongly convex and Lipschtiz continuous loss function $F_z, \forall z \in Z$, to provide guidelines for practitioners to employ SGFusion in real-world applications. The key result is that for a particular zone z, with a careful learning rate decaying process, SGFusion converges to the global minima of zone z with the rate of $\mathcal{O}(\log(T)/T)$, where T is the number of updating steps. Furthermore, our analysis highlights the impact of the non-IID data property among the zones on the convergence of SGFusion and how SGFusion remedies this impact to enhance the model's utility. To do so, firstly, we consider the following assumptions:

Assumption 1. *$F_z, \forall z \in Z$ is μ-strongly convex, we have $F_z(\theta') \geq F_z(\theta) + (\theta' - \theta)^\top \nabla_{\theta'} F_z(\theta') + \frac{\mu}{2}\|\theta' - \theta\|_2^2, \forall \theta', \theta$.*

Assumption 2. *$F_z, \forall z \in Z$ is G-Lipschitz, such that $\|\nabla_\theta F_z(\theta)\|_2 \leq G$.*

Assumption 3. *There exists a constant τ such that $\|\theta_z^* - \theta_{z'}^*\|_2 \leq \tau, \forall z, z' \in Z$, where θ_z^* is the parameter at the global minimum for zone z.*

These assumptions are typical in providing convergence analysis for FL algorithms in the previous works [29]–[31]. Moreover, they are practically common for many ML models, such as linear regression, logistic regression, and simple neural networks [32]. Given these assumptions, we can establish the convergence rate of θ_z through T updating steps, learned by SGFusion, as follows:

Theorem 1. *Let θ_z^T be the output of Alg. 2. If learning rate $\eta_t = \frac{1}{\mu t}$ and Assumption 1 - 3 are satisfied, then the excessive risk $\mathbb{E}[F_z(\theta_z^T)] - F_z(\theta_z^*)$ is bounded by:*

$$\mathbb{E}\Big[(F_z(\theta_z^T)\Big] - F_z(\theta_z^*) \leq \frac{10G^2}{\mu T} + \frac{16G^2}{\mu T}(1 + \log(\frac{T}{2})) + \frac{\bar{G}}{2\mu}\frac{1 + \log(T)}{T} + \frac{3G\tau}{2}\sum_{k \neq i} p_{z,z'}, \quad (14)$$

where $\bar{G} = G^2\big[1 + 2\sum_{z' \neq z} p_{z,z'} + \sum_{z' \neq z} p_{z',z}(1 - p_{z,z'}) + \big(\sum_{z' \neq z} p_{z,z'}\big)^2\big]$, $p_{z,z'}$ is the probability to sample zone z' for the updating process of zone z given the dendrogram $\mathcal{T}$, and the expectation is over the randomness of SGFusion.

Proof. By the updating process of SGFusion, we have that:

$$\theta_z^{t+1} = \theta_z^t - \eta_t\Big[\nabla_{\theta_z^t} F_z + \sum_{z' \neq z} a_{z,z'}\lambda_{z,z'}\nabla_{\theta_z^t} F_{z'}\Big],$$

where $a_{z,z'} \sim Bernoulli(p_{z,z'})$ with $p_{z,z'}$ is the probability to sample zone z' for the updating process of zone z. Let us denote $g_z^t = \nabla_{\theta_z^t} F_z + \sum_{z' \neq z} a_{z,z'}\lambda_{z,z'}\nabla_{\theta_z^t} F_{z'}$, we can expand:

$$\begin{aligned}\|\theta_z^{t+1} - \theta\|_2^2 &= \|\theta_z^t - \eta_t g_z^t - \theta\|_2^2 \\ &= \|\theta_z^t - \theta\|_2^2 - 2\eta_t\langle g_z^t, \theta_z^t - \theta\rangle + \eta_t^2\|g_z^t\|_2^2.\end{aligned}$$

By re-arranging the equations, we have:

$$\begin{aligned}\langle g_z^t, \theta_z^t - \theta\rangle &= \frac{1}{2\eta_t}(\|\theta_z^t - \theta\|_2^2 - \|\theta_z^{t+1} - \theta\|_2^2) + \frac{\eta_t}{2}\|g_z^t\|_2^2 \\ &= \frac{1}{2\eta_t}A_1 + \frac{\eta_t}{2}A_2. \quad (15)\end{aligned}$$

We can bound the expectation of A_2 as follows:

$$\begin{aligned}A_2 &= \Big\|\nabla_{\theta_z^t} F_z + \sum_{z' \neq z} a_{z,z'}\lambda_{z,z'}\nabla_{\theta_z^t} F_{z'}\Big\|_2^2 \\ &\leq \Big[\|\nabla_{\theta_z^t} F_z\|_2 + \sum_{z' \neq z} a_{z,z'}\lambda_{z,z'}\|\nabla_{\theta_z^t} F_{z'}\|_2\Big]^2 \quad (16) \\ &\leq G^2\Big[1 + \sum_{z' \neq z} a_{z,z'}\Big]^2,\end{aligned}$$

where the second inequality is due to the Assumption 2, and $\lambda_{z,z'} \leq 1, \forall z, z'$. Therefore, in expectation, we have:

$$\begin{aligned}&\mathbb{E}(A_2) \leq G^2\mathbb{E}\Big[1 + \sum_{z' \neq z} a_{z,z'}\Big]^2 \\ &= G^2\Big[1 + 2\mathbb{E}\Big(\sum_{z' \neq z} a_{z,z'}\Big) + \mathbb{E}\Big[\Big(\sum_{z' \neq z} a_{z,z'}\Big)^2\Big]\Big] \\ &= G^2\Big[1 + 2\mathbb{E}\Big(\sum_{z' \neq z} a_{z,z'}\Big) + Var\Big(\sum_{z' \neq z} a_{z,z'}\Big) + \mathbb{E}\Big(\sum_{z' \neq z} a_{z,z'}\Big)^2\Big] \\ &= G^2\Big[1 + 2\sum_{z' \neq z} p_{z,z'} + \sum_{z' \neq z} p_{z,z'}(1 - p_{z,z'}) + \Big(\sum_{z' \neq z'} p_{z,z'}\Big)^2\Big] = \bar{G},\end{aligned}$$

where $p_{z,z'}, \forall z, z'$ are fixed given a HRG. Furthermore, we can observe that:

$$\begin{aligned}\langle g_z^t, \theta_z^t - \theta\rangle &= \langle \nabla_{\theta_z^t} F_z, \theta_z^t - \theta\rangle \\ &\quad + \sum_{z' \neq z} a_{z,z'}\lambda_{z,z'}\langle \nabla_{\theta_z^t} F_{z'}, \theta_z^t - \theta\rangle.\end{aligned}$$

By the convexity of $F_z, \forall z$, we can have that:

$$\begin{aligned}\langle \nabla_{\theta_z^t} F_z, \theta_z^t - \theta \rangle &\geq F_z(\theta_z^t) - F_z(\theta) \\ \langle \nabla_{\theta_z^t} F_{z'}, \theta_z^t - \theta \rangle &\geq F_{z'}(\theta_z^t) - F_{z'}(\theta) \\ &= F_{z'}(\theta_z^t) + F_{z'}(\theta_{z'}^*) - F_{z'}(\theta_{z'}^*) - F_{z'}(\theta) \\ &\geq -|F_{z'}(\theta_{z'}^*) - F_{z'}(\theta)| \geq -G\|\theta_{z'}^* - \theta\|_2.\end{aligned}$$

Therefore, putting to Eq. (15), it follows that:

$$\begin{aligned}F_z(\theta_z^t) - F_z(\theta) \leq \frac{1}{2\eta_t} A_1 + \frac{\eta_t}{2} A_2 \\ + G \sum_{z' \neq z} a_{z,z'} \lambda_{z,z'} \|\theta_{z'}^* - \theta\|_2.\end{aligned}$$

Let v be an arbitrary element in $\{1, \ldots, \lfloor T/2 \rfloor\}$. Summing over all over $t = T - v, \ldots, T$, setting $\eta_t = \frac{1}{\mu t}$, and taking expectation on both sides, we have:

$$\begin{aligned}&\mathbb{E}\Big[\sum_{t=T-v}^{T} (F_z(\theta_z^t) - F_z(\theta))\Big] \leq \frac{\mu(T-v)}{2} \mathbb{E}(\|\theta_z^{T-v} - \theta\|_2^2) \\ &+ \frac{\mu}{2} \sum_{t=T-v+1}^{T} \mathbb{E}(\|\theta_z^t - \theta\|_2^2) + \frac{\bar{G}}{2\mu} \sum_{t=T-v}^{T} \frac{1}{t} \\ &+ G(T-u+1)\mathbb{E}\Big[\sum_{k \neq i} a_{z,z'} \lambda_{z,z'} \|\theta_{z'}^* - \theta\|_2\Big].\end{aligned}$$

By the analysis from Theorem 1 from Shamir et al. [33] for strongly convex function and replace θ by θ_z^*, we have:

$$\begin{aligned}&\mathbb{E}\Big[(F_z(\theta_z^T)\Big] - F_i(\theta_z^*)) \leq \frac{10G^2}{\mu T} + \frac{16G^2}{\mu T}(1 + \log(\frac{T}{2})) \\ &+ \frac{\bar{G}}{2\mu} \frac{1 + \log(T)}{T} + \frac{3G}{2} \mathbb{E}\Big[\sum_{k \neq i} a_{z,z'} \lambda z, z' \|\theta_{z'}^* - \theta_z^*\|_2\Big].\end{aligned}$$

By the Assumption 3, we have:

$$\begin{aligned}&\mathbb{E}\Big[(F_z(\theta_z^T)\Big] - F_z(\theta_z^*)) \leq \frac{10G^2}{\mu T} + \frac{16G^2}{\mu T}(1 + \log(\frac{T}{2})) \\ &+ \frac{\bar{G}}{2\mu} \frac{1 + \log(T)}{T} + \frac{3G\tau}{2} \sum_{k \neq i} p_{z,z'},\end{aligned}$$

which concludes the proof. □

As $T \to \infty$, we can induce from Eq. (14) that SGFusion converges to the global minima of each zone $z \in Z$ with the rate of $\mathcal{O}(\log(T)/T)$. From the last term of Eq. (14), we see that even if $T \to \infty$, the performance of SGFusion is limited by the non-IID property among the zones quantified by τ, since $\tau \to \infty$ will enlarge the expected excessive risk. This impact of τ is consistent with the theoretical and empirical results of previous works [29]. However, focusing on the last term of Eq. (14), it also highlights how SGFusion remedies the non-IID problem from the theoretical point of view. Specifically, as $\tau \to \infty$, which means we have more diverse data distributions among different zones, the $p_{z,z'}, \forall z, z' \in Z$ will decrease due to the normalization in Line 10, Alg. 1. Hence, SGFusion decreases last term's value in a heavily non-IID setting, resulting in a better model utility for each zone.

B. Complexity Analysis

This section analyzes the complexity of SGFusion in HRG sampling and SGFusion's training processes. For the HRG sampling, given Z zones, there are $|Z|(|Z|-1)/2$ pairs of zones for the fully connected graph G. Thus, the computation complexity to construct the HRG is $O(|Z|^2)$ because we need to compute the distance of each pairs. Then, given $|n_{L_r}|$ and $|n_{R_r}|$ are the numbers of zones in the left and the right subtrees of the dendrogram $\mathcal{T}$, it takes $O(n_{L_r} * n_{R_r})$ to compute the utility loss $\mathcal{L}(\mathcal{T})$ for one MCMC step. By applying the Cauchy-Schwarz inequality, we have that:

$$n_{L_r} n_{R_r} \leq \frac{(n_{L_r} + n_{R_r})^2}{4} \leq \frac{|Z|^2}{4}. \tag{17}$$

Therefore, it requires $O(M|Z|^2)$ in the worst case for the dendrogram $\mathcal{T}$ to converge, where M is the number of MCMC steps until the convergence of $\mathcal{T}$. Then, to construct the probabilistic (binary) dendrograms, it takes $O(\log|Z|)$ to traverse through the depth of the dendrogram. Hence, it takes $O(|Z|\log|Z|)$ to get the p_r for all the zones. Therefore, the computation complexity of the HRG sampling process scales by $O(|Z|^2 + M|Z|^2 + |Z|\log|Z|) \approx O(M|Z|^2)$. However, this process is only executed once before the SGFusion training. Furthermore, the number of zones $|Z|$ is generally small (less than 40), resulting in low computation cost.

Regarding to the SGFusion training process, we can compute the gradient update for each zone in parallel. Therefore, we consider the computational complexity of a single zone z as follows. In a training round t, the computational complexity to compute the gradient update for the zone's model θ_z is $O(\mathcal{N}(z,t)) \approx O(|Z|)$, where $\mathcal{N}(z,t)$ is the number of neighboring zones which share the gradients with the zone z. Furthermore, the training process is executed with T updating steps. Thus, the computational complexity training process is scaled by $O(T|Z|)$, which is linear to the number of zones $|Z|$. As a result, SGFusion remains computationally scalable in the real-world scenario.

V. Experiments and Evaluation

We conduct extensive experiments using a real-world dataset collected across six countries to evaluate the performance of SGFusion in comparison with state-of-the-art baselines, focusing on the following aspects: **(1)** Assessing zone-FL model utility enhanced by SGFusion; **(2)** Evaluating system scalability of SGFusion through its convergence rate; and **(3)** Understanding the contribution of each component in SGFusion to the overall utility.

Datasets. We use the heart rate prediction (HRP) dataset [28], consisting of approximately $168,000$ workout records of 956 users collected from 33 countries to evaluate SGFusion. Similar to [7], we select the top 6 countries with more than 10 zones having good numbers of data samples, i.e., more than 450 data samples per zone on average, in our experiment. HRP is an outstanding dataset for our study, and it has sufficient users and geographical information across multiple countries. The availability of such datasets is limited in the real world.

TABLE I: Breakdown of the HRP dataset for top six countries: Norway, Spain, US, Thailand, France, and Poland.

	# users	# samples	# zones	avg # samples/zone
Norway	48	5,902	13	454.00
Spain	110	9,609	13	739.15
US	99	14,774	32	461.69
Thailand	105	10,970	23	476.96
France	67	8,094	18	449.67
Poland	205	16,907	16	461.69

Models and Metrics. We leverage a Long Short-Term Memory (LSTM) model [28] to forecast the heart rate from the input features, such as workout altitude, distance, and elapsed time (or speed). We use the Root Mean Square Error (RMSE) as the main metric to evaluate the model utility. The lower the value of RMSE, the better the model utility.

Established Baselines. We consider a variety of baselines: **(1)** The classical **FedAvg** [1]; **(2)** Geographical FL approaches, including Static Geographical FL (**SGeoFL**) and Deterministic Zone Gradient Diffusion (**D-ZGD**) [7]; **(3) IFCA**, an iterative federated clustering algorithm [3]; **(4)** A multi-center FL approach Stochastic Expectation Maximization FL (**FedSEM**) [5]; and **(5)** A sampling FL mechanism **FedDELTA** [9] and its variant **DELTA-Z** proposed to adapt DELTA to geographical zones. FedAvg is the traditional FL setting, where all users jointly train a global FL model. In SGeoFL, users are geographically separated into zones, and every zone trains its own zone-FL model independently without gradient fusing with other zones. D-ZGD is the state-of-the-art training algorithm with self-attention following Eq. 3 given geographical zones. IFCA is a clustering-based FL in which cluster identities of user and model parameters are optimized via a gradient descent process. We apply IFCA on top of each country, where users are distributed and partitioned into clusters without considering any geographical location. FedSEM utilizes an expectation-maximization framework to optimize the client clusters during the FL process. We also adapt FedSEM to the countries' models. DELTA is an FL sampling mechanism in which users are selected at each training round to reduce the diversity of their local gradients and variance, improving the learning process. FedDELTA is an FL approach that utilizes a state-of-the-art sampling mechanism, DELTA, to boost the model performance. Also, since DELTA is not originally designed for geographical zones, we adapt it to DELTA-Z by letting DELTA treat the zone z's model as a central model jointly trained by all users from sampled zones and the zone z. For a fair comparison, we assign a similar number of zones with D-ZGD for the zone sampling process of DELTA-Z.

Variants of SGFusion. We consider a variant of SGFusion, called χ-**SGFusion**, in which every zone $z \in Z$ samples the same number of zones χ in their gradient fusion across training rounds. Also, we include **top-k-SGFusion**, in which every zone z uses top-k most similar zones $\{z'\}$, i.e., smallest distance $d(\mathcal{Y}_z, \mathcal{Y}_{z'})$, in its gradient fusion across training rounds. The goals of considering these variants of SGFusion are: **(1)** For a fair comparison with D-ZGD, we set χ for every zone z to be the same with the number of neighboring zones of the zone z used in D-ZGD; and **(2)** Evaluating the effect of stochastic gradient fusion compared with deterministic gradient fusion using either a fixed number of sampled zones or top-k most similar zones with different values of k.

TABLE II: SGFusion vs. D-ZGD and DELTA-Z regarding the number of zones with higher model utility.

	D-ZGD	SGFusion	(%) Gain	DELTA-Z	SGFusion	(%) Gain
Norway	5	8	60.00%	4	9	125.00%
Spain	5	8	60.00%	4	9	125.00%
US	10	22	120.00%	8	24	200.00%
Thailand	8	15	87.50%	11	12	9.09%
France	6	12	100%	7	11	57.14%
Poland	4	12	200%	4	12	200.00%

TABLE III: χ-SGFusion vs. D-ZGD and DELTA-Z regarding the number of zones with higher model utility.

	D-ZGD	χ-SGFusion	(%) Gain	DELTA-Z	χ-SGFusion	(%) Gain
Norway	6	7	16.67%	4	9	125.00%
Spain	5	8	60.00%	3	10	233.33%
US	11	21	90.91%	7	11	57.14%
Thailand	10	13	30.00%	10	13	30.00%
France	4	14	250.00%	7	11	57.14%
Poland	7	9	28.57%	8	8	0.00%

A. Utility and Scalability

Zone and Country Level Utility. Among all the baselines, D-ZGD and DELTA-Z achieve the best performance (Figure 7). Tables II and III present the utility of SGFusion compared with these best baselines at the zone-level FL model. SGFusion and χ-SGFusion achieve significantly better model utility on most zone-FL models than D-ZGD and DELTA-Z across all countries. For instance, in Poland, SGFusion achieves better model utility in 12 zones compared with 4 zones of D-ZGD, registering a 200% improvement. Similar results are observed from other countries and with other baselines, indicating the sharp enhancement of the model utility by the SGFusion at the level of zone-FL models. Across 115 zones in six countries, more than double the number of zones benefit from SGFusion, i.e., 77 zones, than the best baselines, i.e., 38 zones.

Although the improvement at the country level (Figure 7) is not as clear as the zone level since each country uses an aggregated model from its zone models, the model utility at the country level can strengthen our observation. SGFusion improves the aggregated model utility across six countries by 3.23% on average compared with existing approaches. Note that χ-SGFusion, which uses the same numbers of sampled zones with D-ZGD, also outperforms D-ZGD across six countries and outperforms DELTA-Z in five countries with a comparable result in Norway.

Convergence Speed. As shown in Figure 8a, SGFusion and χ-SGFusion have a similar convergence speed compared with D-ZGD. The key reason for this result is that SGFusion utilizes relatively small numbers of sampled zones to train a specific zone-FL model on average compared with D-ZGD and DELTA-Z (Figure 8b). In fact, at a training round, the average numbers of sampled zones in SGFusion are smaller

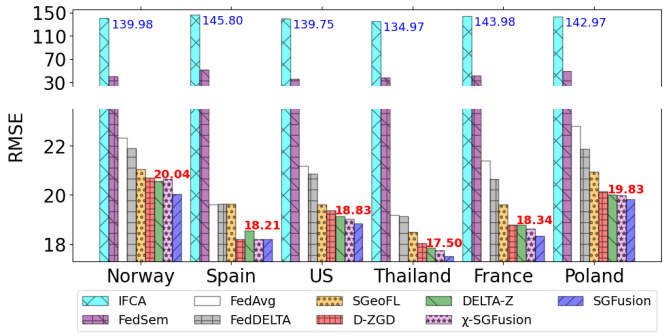
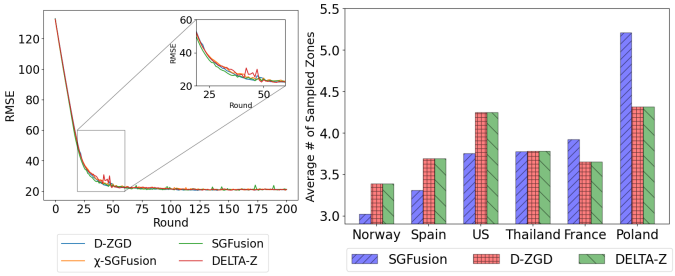


Fig. 7: Model utility across countries (smaller, the better).



(a) Learning curve. (b) Average number of sampled zones.

Fig. 8: Learning curves of the country-level FL models using data collected from Norway (best view in color).

than the ones in D-ZGD and DELTA-Z across 4 over 6 countries and are comparable in the remaining countries. To shed light on why using smaller sampled zones for gradient fusion in SGFusion enables us to avoid negative impacts on convergence speed and system scalability, while still resulting in better model utility, we conduct a **homophily data analysis** on these sampled zones. This experiment studies the average homophily [34] across the zones $z \in Z$ and across T updating steps, quantified as:

$$\frac{1}{T}\sum_{t=1}^{T}\Big[\frac{1}{|Z|}\sum_{z\in Z}\Big(\frac{1}{|\mathcal{N}(z,t)|}\sum_{z'\in\mathcal{N}(z,t)} d(\mathcal{Y}_z,\mathcal{Y}_{z'})\Big)\Big], \quad (18)$$

where $d(\cdot,\cdot)$ is the metric measuring the data label distributions between zones z and z', e.g., Euclidean distance [25]. Intuitively, the lower average homophily, the more similar the label distribution of a zone z and the label distribution of its sampled zones. SGFusion achieves a lower value of average homophily across six countries compared with DELTA-Z and D-ZGD (Figure 9). *The results highlight that SGFusion obtains a better set of sampled zones for gradient fusion, one key property contributing to the improvement in the model utility of SGFusion without affecting system scalability, demonstrated through its convergence speed.*

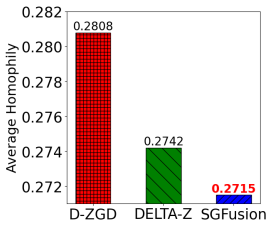


Fig. 9: Average homophily of zones across six countries.

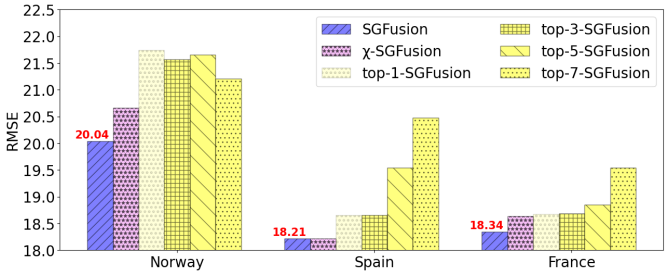


Fig. 10: Deterministic vs. stochastic gradient fusion.

B. Stochastic vs. Deterministic

In this experiment, we investigate the benefit of stochastic gradient fusion by comparing SGFusion with the top-k-SGFusion where k varies from 1 to 7, and with χ-SGFusion. The top-k-SGFusion is a deterministic gradient fusion approach using the top-k most similar zones to fuse gradients. Meanwhile, χ-SGFusion is a partially stochastic gradient fusion algorithm, in which the number of sampled zones for every zone z is deterministic while the sampled zones can change across training rounds. Therefore, using the top-k-SGFusion and χ-SGFusion offers a comprehensive evaluation of the stochastic gradient fusion effect in SGFusion.

In Figure 10, SGFusion notably outperforms the top-k-SGFusion with $k \in [1,7]$ and χ-SGFusion. The reason is that using either deterministic values of top-k or deterministic numbers of sampled zones does not offer a good balance between obtaining sufficient knowledge fused from sampled zones and mitigating the increase of discrepancy among fused gradients when the number of sampled zones increases. The stochastic zone sampling approach remedies this problem by enabling knowledge fusion and sharing among zones, such that *"more similar"* zones have *"higher probabilities"* of sharing gradients with *"larger attention weights"* at a training round of a zone-FL model. Hence, SGFusion reduces better the discrepancy among fused gradients while providing sufficient knowledge from sampled zones to improve zone-FL models.

VI. Discussion

The granularity of zones is an underexplored aspect in the current setting of SGFusion. Zones must not be too large or too small to encompass a sufficient number of users to maintain reliable and high model utility while reducing computational and operational costs. At the same time, the zones must capture localized behavioral differences, such as unique mobility patterns or resource consumption trends. One solution to this problem is to collect user mobility statistics at the edge nodes to identify common mobility patterns and then define the zones based on these patterns.

In addition, the edge infrastructure is another consideration for deploying SGFusion in real-world scenarios. In the current setting, each zone is associated with one edge device, which manages the mobile devices within its boundaries. Assuming the wireless telecom operators will deploy edge devices at scale in the near future, practical deployment of SGFusion may

have several edge devices within one zone. A direct solution is to allow the mobile devices to communicate with any edge devices, but only one edge device must be responsible for the zone. The practical deployment will need communication protocols between the edge devices to ensure the correct functionality of SGFusion. We plan to explore these open research questions in our future work towards real-world deployment of SGFusion.

VII. Conclusions

This paper presented **SGFusion**, a novel FL training algorithm for geographical zones, that models local data label distribution-based correlations among geographical zones as hierarchical and probabilistic random graphs, optimized by MCMC sampling. At each step, every zone samples a set of zones from its associated probabilistic dendrogram to fuse its local gradient with shared gradients from these zones. SGFusion enables knowledge fusion and sharing among zones in a probabilistic and stochastic gradient fusion process with self-attention weights. Theoretical and empirical results show that models trained with SGFusion converge with upper-bounded expected errors and remarkable better model utility without notable cost in system scalability compared with baselines.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*. PMLR, 2017.

[2] T. Zhang and et al., "Federated learning for the internet of things: Applications, challenges, and opportunities," *IoT*, vol. 5, no. 1, 2022.

[3] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," in *NeurIPS*, vol. 33. Curran Associates, Inc., 2020.

[4] Y. Ruan and C. Joe-Wong, "Fedsoft: Soft clustered federated learning with proximal local updating," in *AAAI*, vol. 36, no. 7, 2022.

[5] G. Long, M. Xie, T. Shen, T. Zhou, X. Wang, and J. Jiang, "Multi-center federated learning: clients clustering for better personalization," *WWW*, vol. 26, no. 1, 2023.

[6] Z. Qu and et al., "On the convergence of multi-server federated learning with overlapping area," *TMC*, vol. 22, no. 11, 2022.

[7] X. Jiang, T. On, N. Phan, H. Mohammadi, V. D. Mayyuri, A. Chen, R. Jin, and C. Borcea, "Zone-based federated learning for mobile sensing data," in *PerCom*. IEEE, 2023.

[8] A. Clauset, C. Moore, and M. E. Newman, "Structural inference of hierarchies in networks," in *ICML workshop on statistical network analysis*. Springer, 2006.

[9] L. Wang, Y. Guo, T. Lin, and X. Tang, "DELTA: Diverse client sampling for fasting federated learning," in *NeurIPS*, 2023.

[10] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *IJCNN*. IEEE, 2020.

[11] A. Ghosh and et al., "An efficient framework for clustered federated learning," *NeurIPS*, vol. 33, 2020.

[12] Y. Li, X. Wang, and L. An, "Hierarchical clustering-based personalized federated learning for robust and fair human activity recognition," *IMWUT*, vol. 7, no. 1, 2023.

[13] M. Morafah, S. Vahidian, W. Wang, and B. Lin, "Flis: Clustered federated learning via inference similarity for non-iid data distribution," *OJCS*, vol. 4, 2023.

[14] C. Li, G. Li, and P. K. Varshney, "Federated learning with soft clustering," *IoT*, vol. 9, no. 10, 2022.

[15] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," 2021. [Online]. Available: https://arxiv.org/abs/1910.06378

[16] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2020. [Online]. Available: https://arxiv.org/abs/1812.06127

[17] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," 2021. [Online]. Available: https://arxiv.org/abs/2102.07623

[18] X. Ouyang, Z. Xie, J. Zhou, G. Xing, and J. Huang, "Clusterfl: A clustering-based federated learning system for human activity recognition," *ACM Trans. Sen. Netw.*, dec 2022.

[19] W. Zhang and et al., "Optimizing federated learning in distributed industrial iot: A multi-agent approach," *JSAC*, vol. 39, no. 12, 2021.

[20] J. Wu and et al., "Topology-aware federated learning in edge computing: A comprehensive survey," *CSUR*, 2023.

[21] Beltrán and et al., "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *COMST*, 2023.

[22] T. Li and et al., "Federated learning: Challenges, methods, and future directions," *SPM*, vol. 37, no. 3, 2020.

[23] M. E. Newman and G. T. Barkema, *Monte Carlo methods in statistical physics*. Clarendon Press, 1999.

[24] C. Dwork and et al., "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, 2014.

[25] B. O'Neill, "Chapter 2 - frame fields," in *Elementary Differential Geometry (Second Edition)*, second edition ed. Academic Press, 2006.

[26] E. F. Krause, "Taxicab geometry," *The Mathematics Teacher*, vol. 66, no. 8, 1973.

[27] S. Theodoridis and K. Koutroumbas, "Chapter 14 - clustering algorithms iii: Schemes based on function optimization," in *Pattern Recognition (Fourth Edition)*, fourth edition ed. Academic Press, 2009.

[28] J. Ni, L. Muhlstein, and J. McAuley, "Modeling heart rate and activity data for personalized fitness recommendation," in *WWW*, ser. WWW '19. ACM, 2019.

[29] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," 2020.

[30] H. Xing, O. Simeone, and S. Bi, "Federated learning over wireless device-to-device networks: Algorithms and convergence analysis," *JSAC*, vol. 39, no. 12, 2021.

[31] X. Wu, F. Huang, Z. Hu, and H. Huang, "Faster adaptive federated learning," in *AAAI*, vol. 37, no. 9, 2023.

[32] M. Pilanci and T. Ergen, "Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks," in *ICML*. PMLR, 2020.

[33] O. Shamir and T. Zhang, "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes," in *ICML*. PMLR, 2013.

[34] K. Z. Khanam, G. Srivastava, and V. Mago, "The homophily principle in social network analysis: A survey," *MTAP*, vol. 82, no. 6, 2023.