Tsinghua Science and Technology
Doi: 10.26599/TST.2026.9010032

Research/Review Article

Adaptive Federated Clustering via Gravitational Dynamics

Guangxi Lu, Lizong Zhang (✉), Chong Mu, and Haoji Zhang

Abstract Federated clustering effectively addresses the Non-IID problem by organizing clients into clusters for the training of personalized models. However, current federated clustering methods often cluster clients based on a single dimension, and fail to simultaneously achieve low computational cost, high accuracy, and strong privacy preservation. To address this problem, this manuscript proposes a novel approach called Gravitational Clustering Federated Learning (GCFL). GCFL treats each client as an object in a latent space, where the position encodes the local model and the mass encodes client importance. By simulating gravitational interactions between clients, GCFL enables adaptive clustering. Extensive experiments on Non-IID datasets validate the effectiveness of GCFL, and comparative analysis with state-of-the-art methods demonstrates that the proposed approach achieves more reasonable clustering and faster convergence.

Keywords Federated learning, Non-IID Problem, Adaptive Client Clustering, Personalized Models, Gravitational Dynamics

1 Introduction

Data privacy is always a key requirement in machine learning. In this background, Federated Learning (FL) has attracted wide interest in mobile applications[1,2], healthcare[3,4], and finance[5,6].

FL is a privacy-preserving distributed learning paradigm. It uses a central server to aggregate client updates, and multiple clients can jointly train a global model without their private data leaving the local devices. Concretely, in each training round, the server selects a subset of clients and sends them the global model. Each client performs several local epochs of stochastic gradient descent on its private dataset to produce and return the model updates. The server then aggregates updates to obtain the next global model. Throughout training, raw data remain on local devices thereby reducing the risk of leakage.

However, FL still struggles with several problems, and data heterogeneity, also called Non-IID, is the primary one.[7] The Non-IID problem means that local data distributions are different across clients, for example, skewed label distributions, unequal sample sizes, and varying class coverage. Non-IID will cause oscillations across training rounds and slow convergence, ultimately resulting in a global model with weak generalization and inconsistent performance across clients[8].

Clustering Federated Learning (CFL) is a common solution to solve the Non-IID problem[9,10]. Compared to FL, which trains a single global model for all clients, CFL groups clients with similar data distributions into clusters, and each cluster will train a personalized model[11]. CFL improves model accuracy by customizing personalized models for each cluster and increases convergence speed since the data distributions in each cluster are similar.

CFL follows three different paths: pre-training clustering, in-training clustering, and post-training clustering. Pre-training clustering[12–14] refers to clustering clients based on information collected before FL. Good pre-training clustering can significantly improve convergence speed. However, since it relies on additional information, it cannot guarantee both privacy protection and clustering performance.[15] In-training clustering[9,16,17] refers to clustering clients during the iterative process. It clusters clients based on gradients and model parameters, thereby ensuring privacy. However, In-training clustering usually requires pre-setting the number of clusters or some specific threshold. These hyperparameters are sensitive to changes in the final result. In-training clustering cannot guarantee both convergence speed and clustering performance. Post-training clustering[18–20] refers to clustering clients after FL, and then all clusters will do FL again to personalize the models. Post-training clustering can typically achieve perfect clustering results in one step. However, post-training clustering is essentially a post-processing method, meaning that the overall speed is much slower than regular

---

• Guangxi Lu is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China. E-mail: glu3@uestc.edu.cn
• Lizong Zhang is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China. E-mail: l.zhang@uestc.edu.cn
• Chong Mu is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China. E-mail: muchong@uestc.edu.cn
• Haoji Zhang is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China. E-mail: hzhang070@e.ntu.edu.sg

Manuscript received: 24-Aug-2025; revised: 17-Dec-2025; accepted: 05-Mar-2026

FL. In summary, CFL struggles to simultaneously achieve low computational cost, high accuracy, and strong privacy preservation. Each approach has its own trade-offs.

To address these issues, this manuscript proposes an adaptive in-training federated clustering method called Gravitational Clustering Federated Learning (GCFL). In GCFL, each client is represented as an object, and GCFL simulates gravitational interactions among objects to drive adaptive clustering without predefining the number of clusters. Specifically, the server anchors local models in a common, comparable embedding space using a small probe set. It also encodes data volume, model performance, and model stability into the ”mass” term. GCFL uses gravitational forces to encourage proximate client objects to coalesce into clusters. Furthermore, GCFL introduces a gravitational shielding mechanism that suppresses cross-cluster attraction to prevent gravitational ”collapse”.

GCFL has three key advantages. First, adaptive clustering driven by gravitational dynamics removes sensitive hyperparameters, such as the number of clusters, leading to more accurate FL results. Second, since GCFL performs clustering during training, it increases convergence speed while preserving the privacy guarantees. Third, clustering relies on both “location” and “mass”, which represent the data volume, probe set performance, and update stability. So GCFL integrates complementary evidence and achieves more reasonable clusters.

The contributions of this manuscript are as follows:

- Adaptive in-training federated clustering: This manuscript introduces GCFL, an in-training clustering framework that models clients as bodies in a latent space, removing the need to predefine the number of clusters or tune sensitive thresholds;
- Mass & Location modeling: GCFL designs a probe set anchoring pipeline to embed clients as the ”location” and define a multi-factor ”mass” term to represent client importance;
- Collapse-prevention mechanisms: GCFL designs gravitational shielding mechanism that decrease long-range attraction, preventing ”collapse” into a single cluster and reducing cross-cluster interference.
- Extensive empirical validation: This manuscript conducts extensive experiments on Non-IID datasets and performs a comparative analysis with state-of-the-art methods, showing that GCFL achieves higher accuracy and faster convergence.

The remainder of the manuscript is divided as follows: Section 2 introduces the prior work on federated clustering. Section 3 introduces the GCFL framework, which includes overall architecture, latent-space anchoring, mass modeling, gravitational clustering, and collapse-prevention mechanisms. Section 4 introduces the experimental setup and results. Section 5 summarizes the contributions and outlines directions for future work.

2 Related Work

Clustering Federated Learning (CFL) augments federated learning by partitioning clients into clusters with similar data distributions or model behavior and then training a personalized model per cluster. It is a common solution to solve the Non-IID problem. Existing CFL methods often follow three paths: pre-training clustering, which forms clusters before training using additional information and can speed early convergence but has privacy issues; in-training clustering, which discovers and clusters clients during training using model updates or gradients and preserves privacy but often depends on sensitive hyperparameters; and post-training clustering, which identifies clusters after FL and then retrains models within clusters, achieving stable partitions at the cost of extra training latency.

2.1 Pre-training Clustering

Side information in Pre-training clustering methods could be severl signals send by client such as : meta-learned signals, which either learning a clustering rule or a fast-adaptation initialization[12,13]; subspace summaries signals, which uploading a few truncated-SVD directions or feature fingerprints[14,21]; cross-client distances signals , which use MPC to calculate and then clustering [22]; and local-center signals, where clients summarize data by local centroids and the server clusters these in a single step[23,24].

Beyond static, one-shot partitioning, recent pre-training clustering can also combine with FL training process. Li et al. [25] propose a drift-aware clustered FL framework. It compares label-distribution vectors to detect the drifted clients. This method preserving inter-cluster homogeneity and improving both accuracy and training time. Helcig et al.[26] proposed FedCCL, a federated clustered continual learning framework, FedCCL use DBSCAN [27] to cluster the client based on static attributes, and then trains with a three-tier global-cluster-local framework.

Pre-training clustering solves Non-IID problem before training begins. It stabilizes early rounds, and speeds up overall training time. Such one-shot partitioning is also modular, and can plugged into standard FL pipelines with low warm-up cost. However, it relies on side information, creating a privacy–utility trade-off and making performance sensitive to the fidelity of those signals.

2.2 In-training Clustering

In-training clustering is the most widely approach in federated clustering. In each round, the server estimates and updates cluster identities, then redefine the cluster group for each client before next FL round. This iteration continues until FL stops, and at also the point that clustering has naturally converged. The In-training clustering supports hard or soft assignments, synchronous or asynchronous framework, and can also combine with differential privacy or other privacy mechanisms.

A representative In-training clustering is IFCA proposed by Avishek et al.[9]. IFCA alternates between estimating client–cluster identities from returned losses and minimizing the loss of each cluster model; assignments and models are updated each round until training ends. However, the stuides show that IFCA is very sensitive to the early-round noise or hyperparameters such as clusters numbers, and hard switch can oscillate easily when clusters overlap or data drift.

To solve this problem, soft-assignment methods were introduced. Yichen et al.[16] propose FedSoft, which learn a mixture weights that softly allocate each client across clusters and jointly optimizes cluster and personalized components. Othmane et al.[17] propose FedEM, it casts CFL as a mixture-of-experts problem with client-specific expert weights and expert-wise aggregation on the server, so achieve soft clusters in FL training round.

Concurrently, another line of In-training Clustering is to change the FL scheduling. Long et al.[28] propose a Multi-center FL, which maintains several global centers and lets each client attach to the nearest center by parameter or update distance; Du et al.[29] propose AICFL, which allows clusters to split or merge over time so that structure adapts as training progresses; Liu et al.[30] propose CASA, which couples clustering with asynchronous aggregation, forming clusters from the stream of arriving updates; Lin et al.[31] propose FedSPD, which groups clients by similarity of gradient or sampling directions to balance accuracy with communication efficiency; Kim et al. [32] propose a gradient-based partitioning, which accumulates gradients and clusters by gradient similarity, with accompanying error bounds; Gu et al. [33] propose LCFL, which defines a loss-based clustering metric suited to nonconvex models.

Recently, in-training clustering have increased robustness by using more advanced training signals. Kim et al.[32] propose CFL-GP, which accumulates client gradients and clusters clients based on gradient similarity, enabling more stable cluster estimation under heterogeneous data. Junbao et al.[34] propose FedCCFA, which leverages client-side classifiers to address distributed concept drift and uses the resulting clusters to guide feature alignment during training.

Despite In-training Clustering has substantial progress, it still share two core limitations. First, these method also rely on some sensitive hyperparameters, thresholds, or stopping rules. whose tuning is fully depends on data distrabute and can induce unstable oscillations under early-round noise or distribution drift. Second, most methods cluster on a single signal, overlooking complementary factors like data volume, update reliability, and temporal stability.

2.3 Post-training Clustering

The typical pipeline of post-training clustering is follows this template: run FL to a stable point, extract features from each client, compute distances in a common space, group clients in a single shot, and then continue with per-cluster retraining or fine-tuning. Post-training clustering is easily extracts the low-noise signals, therefore forming more reasonable clusters compared with other methods. Post-training Clustering reduces cross-client heterogeneity, stabilizes subsequent optimization, and improves personalization quality.

The most classical post-training method is CFL[18]. CFL first trains an model to their near-stationary point, and computes cosine similarities between client gradients. It then uses recursive bisection to split clients until within-cluster consistency is reached. CFL is simple but follows a "train–cluster–retrain" pipeline, which leading to high communication and computation costs.

Besides CFL, recent research tries to improve the efficiency, robustness, and deployability of post-training clustering: Zhang et al.[35] propose a communication-efficient CFL, which defines a new model distance and refines post-hoc bisection. The communication-efficient CFL reduces per-round communication, addressing the cost issue; Vardhan et al.[36] further strengthen the distance metric to achieve more robust post-processing splits; Zuziak et al.[20] propose a one-shot CFL, which detects the earliest near-stationary point; therefore, it has a lower cost; Xue et al.[37] propose DAG-ACFL, which combines CFL with an asynchronous DAG pipeline; Briggs et al.[19] propose FL+HC, which first designs a hierarchical clustering pipeline for local updates; Guo et al.[38] propose HCFL, which analyzes hierarchical grouping and merge–split strategies. HCFL also explores similarity measures beyond cosine for more stable clustering; Liu et al.[39] propose a hierarchical layered CFL for multi-tier networks, which applies hierarchical clustering to reduce latency; Jhunjhunwala et al.[40] propose FedFisher, which leverages the Fisher information matrix–based knowledge distillation to reduce retraining cost; Zhang et al.[41] propose SoFL, which employs self-organizing maps to adapt the number of clusters and decouples clustering from FL into a two-stage pipeline. The post-training clustering achieve high-quality cluster result, but its core scheme remains "train-cluster-retrain". One-shot or hierarchical variants can only reduce the clustering step but cannot remove the retrain stage. Therefore, the latency, communication, and computational complexity of post-training clustering are inevitably higher.

3 Methods

3.1 Problem Formulation

Federated clustering divides the client set $C = \{C_1, C_2, \ldots, C_n\}$ into m disjoint and non-overlapping clusters $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_m\}$. For clients $C_a, C_b \in \mathcal{G}_i$, the data difference satisfies $\Delta(D_a, D_b) < \xi_{intra}$. For clients $C_a \in \mathcal{G}_i$ and $C_b \in \mathcal{G}_j$, the data difference satisfies $\Delta(D_a, D_b) > \xi_{inter}$, with $\xi_{inter} > \xi_{intra}$.

The goal of clustering is to learn a personalized model θ_i for each cluster $\mathcal{G}_i$, such that the model better fits the local data of the clients. Specifically, for each client $C_a \in \mathcal{G}_i$, the objective is to ensure that the expected loss of θ_i is lower than the expected loss of single global model θ_{global}, i.e.,

$$\mathbb{E}[\ell(\theta_i; D_i)] < \mathbb{E}[\ell(\theta_{global}; D_i)], \tag{1}$$

where D_i representing the data across all clients within the cluster $\mathcal{G}_i$.

3.2 Gravitational Clustering Federated Learning Framework

Gravitational Clustering Federated Learning (GCFL) is an in-training clustered federated learning framework that forms client clusters and performs per-cluster aggregation. Given the client set $C = \{C_1, \ldots, C_n\}$, each communication round t the server maintains a dynamically evolving cluster set $\mathcal{G}^t = \{\mathcal{G}_1^t, \ldots, \mathcal{G}_{m_t}^t\}$ with an adaptively varying number of clusters m_t. Therefore, the server aggregate and maintains a personalized model set $\theta^t = \{\theta_1^t, \ldots, \theta_{m_t}^t\}$.

The overall framework of GCFL is shown in Fig. 1. Fig. 1(a) shows how GCFL cold-starts, while Fig. 1(b) shows the full federated training loop. For cold-starts, the server first initializes a single model θ_{global}^0, and server broadcasts the global model θ_{global}^t at each early round t. Then , the client trains and then update the local model θ_a^t. The server aggregates the local models and updates the global model:

$$\theta_{global}^{t+1} = \sum_{a=1}^{n} \frac{N_a}{N} \theta_a^{t+1} \tag{2}$$

where N is all data volume, N_a is local data volume. Once the global model becomes stable, GCFL will start clustering. Specifically, GCFL check the following criterion:

$$\frac{\|\theta_{global}^{t+1} - \theta_{global}^t\|_2}{\|\theta_{global}^t\|_2 + \varepsilon} \leqslant \delta_{init}, \tag{3}$$

where $\varepsilon > 0$ is a small constant to avoid division by zero.

For clustering training phase, the server first use gravitational method to cluster all client. Therefor at round t, the server holds a cluster $\mathcal{G}^t = \{\mathcal{G}_1^t, \ldots, \mathcal{G}_{m_t}^t\}$. parameters are then aggregated within each cluster to get personalized model θ_i^t

$$\theta_i^t = \sum_{C_b \in \mathcal{G}_i^t} \frac{N_b}{N_i} \theta_b^t, \tag{4}$$

where N_b is local data volume of client C_b, and N_i is the total data volume in cluster $\mathcal{G}_i^t$. The server then broadcasts the personalized model θ_i^t to all cluster members client. Each client who get θ_i^t, runs a few local SGD steps on local data to obtain a local model θ_a^{t+1}

$$\theta_a^{t+1} = \theta_i^t - \eta \nabla L_a\left(\theta_i^t\right), \tag{5}$$

where $\nabla L_a\left(\theta_i^t\right)$ is the local gradient, η is learning rate.

3.3 Gravitational Clustering

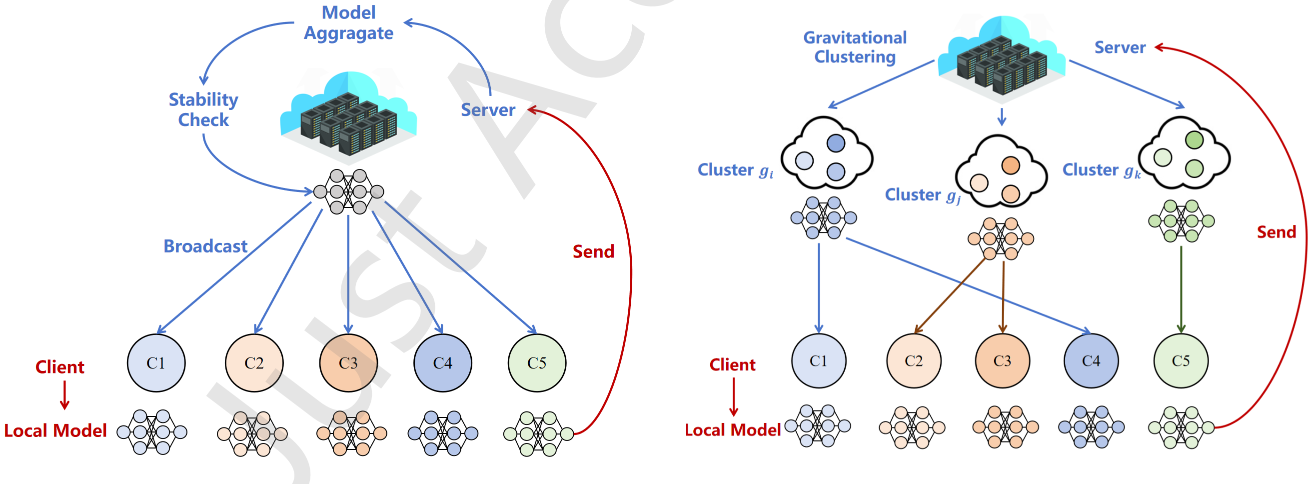
To address the training oscillations, aggregation mismatch, and hyperparameters sensitivity problem in traditional CFL, GCFL introduces a gravity-based clustering method. The core idea borrows from the mutual attraction in the physical world: each client is modeled as a "object " in a latent space, where the embedding of its local model serves as "location", and the client importance is encoded as its "mass". Intuitively, more important clients exert stronger attraction on similar neighbors and are less susceptible to perturbations from small, dissimilar clients. Therefore the clients attract each other with a force strength increases with mass and decays with distance. Local attraction drives nearby clients to group up adaptively, forming stable cluster during federated training.

3.3.1 Location Anchoring Phase

In location anchoring phase, server in GCFL first need a small probe set P. In FL, the server theoretically knows the label space. Therefor, P can be bulid as a tiny, label balanced micro dataset covering all labels, it can be few public or crowdsourced examples per label, or a synthetic set generated from public sources. With this probe set, GCFL calculate

Table 1 Annotation Table

Symbol	Description
C	Set of all clients.
n	Total number of clients.
$\mathcal{G}$	Set of clusters formed over clients.
m	Number of clusters.
t	Communication round index in federated training.
$D_a,\ D_b$	Local datasets of clients C_a and C_b.
$\Delta(D_a, D_b)$	A divergence measure between client datasets (smaller means more similar).
θ^t_{global}	Global model maintained by the server at round t.
θ^t_i	Cluster-personalized model for cluster $\mathcal{G}_i$ at round t.
$\nabla L_a(\cdot)$	Gradient of client C_a's local loss.
N	Total data volume.
P	Probe dataset
$g_{a,k}(x)$	The k-th logit output by model θ_a on sample x.
$q_a(x, y)$	Softmax confidence of client C_a on the true class y for sample (x, y).
$r_{a,k}$	Average confidence of client C_a on class k over P_k.
z_a	Location vector of client C_a in the latent space.
ν_a	Data-volume factor indicating the relative size of client C_a's dataset.
κ_a	Reliability factor reflecting C_a's participation ratio in recent rounds.
ρ_a	Confidence factor reflecting C_a's average probe set confidence.
m_a	Mass factor to client C_a.
d_{ab}	Euclidean distance between clients a and b in the location space.
A_{ab}	Attraction strength between clients a and b.
S_a	Total attraction of client a to all other clients.
$\widetilde{A}_{ab}$	Symmetrically normalized attraction between clients a and b.



(a) (b)

Fig. 1 Gravitational Clustering Federated Learning Framework

local model confidence. Specifically, for any client C_a with model parameters θ_a, define the confidence as

$$q_a(x, y) = \frac{\exp\left(g_{a,y}(x)\right)}{\sum_{k=1}^{|y|} \exp\left(g_{a,k}(x)\right)} \tag{6}$$

where $(x, y) \in P$ is a sample with ground-truth label y, $|y|$ is the number of classes, $(g_{a,1}(x), \ldots, g_{a,M}(x))$ is the logits produced by the model θ_a on data sample x.

GCFL then aggregate the confidence by class to get class-wise confidence

$$r_{a,k} = \frac{1}{|P_k|} \sum_{(x,y)\in P_k} q_a(x, y) \tag{7}$$

where Let P_k is the probe subset with true label k. To calculate the location of each client, GCFL stack all class-wise confidence $[r_{a,1}, \ldots, r_{a,M}]$ as location vector r_a. GCFL also normalize this location vector to removes scale differences while preserving the relative per-class pattern. Therefore the final location vector z_a is calculate by

$$z_a = \frac{r_a}{\|r_a\|_2}, \tag{8}$$

3.3.2 *Mass Modeling Phase*

In the mass modeling phase, GCFL assigns each client with a non-negative scalar mass number. This mass encodes the relative importance of the client. Specifically, mass reflects the reliability and influence of each client. The client with more data, high reliable, more confident predictions should have a stronger attraction to other client. while client with small and noisy data should have weaker attraction.

Formally, The first mass factor is data volume. Clients with more samples are more representative. To reflect this, GCFL define a normalized data volume factor as

$$\nu_a = \frac{N_a}{\max N}, \tag{9}$$

where N_a is the data volume of client C_a, and $\max N$ is the max data volume of all client.

The second factor is Client reliability, which reflects whether this client participates in federated learning. Specifically, reliability factor define as

$$\kappa_a = \frac{T_a}{T}. \tag{10}$$

where T_a denote the number of rounds in which client C_a successfully uploads updates within a recent sliding window of length T.

The third factor is confidence, which evaluates accuracy of model performance for each client, Specifically, confidence factor define as

$$\rho_a = \frac{1}{|P|} \sum_{(x,y)\in P} q_a(x, y), \tag{11}$$

where $q_a(x, y)$ is the softmax score of client C_a on sample (x, y).

Therefore, the mass of client C_a is calculated as

$$m_a = \alpha\nu_a + \beta\kappa_a + \gamma\rho_a. \tag{12}$$

3.3.3 *Gravitational Phase*

In gravitational phase, GCFL adaptive cluster each client based on their mass–location representation (z_a, m_a). Specifically, GCFL calculates the mutual force of each client and then takes the strong force to group as clusters. To prevent the collapse, GCFL uses distance scale to suppress the long-range attraction, and use the symmetric degree normalization to down-weight global hubs. Therefore, GCFL achieve a stable clusters without presetting the number of clusters or hand-tuning thresholds.

The above section has calculate the location $z_a \in \mathbb{R}^M$ and the mass m_a for each client C_a, GCFL first define distances of client a and b as

$$d_{ab} = \|z_a - z_b\|_2 \tag{13}$$

According to the gravitational principle in natural world, the force between two objects should be proportional to their masses and inversely proportional to their distance. In this case, any attraction force between client pair (C_a, C_b) in FL can be calculate as

$$A_{ab} = \frac{m_a m_b}{d_{ab} + \zeta}, \tag{14}$$

where $\zeta > 0$ is a small stabilizer set to prevents the force blow-ups when d_{ab} is very small.

To suppress the attraction force between high-mass and long-range clients, GCFL introduce a shielding factor to calculate the attraction as:

$$A_{ab} = \frac{m_a m_b}{d_{ab} + \zeta} \exp^{-\left(\frac{d_{ab}}{\lambda}\right)^2} \tag{15}$$

Where λ is a data-driven distance scale, which is the median of all distances.

To further suppress the client in a dense region to link almost all others, GCFL applies symmetric normalization as:

$$S_a = \sum_{b\neq a} A_{ab}, \quad \widetilde{A}_{ab} = \frac{A_{ab}}{\sqrt{S_a S_b}} \tag{16}$$

Where, S_a is the total attraction of node a. This normalization preserves symmetry where $\widetilde{A}_{ab} = \widetilde{A}_{ba}$, and down-weights the hub effects.

Then GCFL will calculate a Otsu thresholding τ[42] to separate strong from weak attractions, and if $\widetilde{A}_{ab} \geqslant \tau$, then group (C_a, C_b), the algorithm of Gravitational cluster is shown as Algorithm 1

4 Experimental

4.1 Datasets

Experiments are conducted on the CIFAR-10 and MNIST datasets. CIFAR-10 contains 60000 32 by 32 RGB images from 10 classes. MNIST contains 70000 28 by 28 grayscale digit images from 10 classes. For both datasets, a small probe set P is randomly sampled before training. In this experiment, the probe set is constructed by uniformly sampling 50 instances per class, resulting in a probe set of 500 samples for both CIFAR-10 and MNIST. To preserve privacy, the probe set is held exclusively by the server and is never assigned to any client. It is only used to compute class-wise confidence and mass modeling. After removing the probe set, the remaining data are partitioned across clients under three data-distribution settings:

- **IID.** Samples are evenly split to each clients, therefore each client receives a balanced subset.

Algorithm 1 GCFL: Gravitational Clustering

```
**Input:** client set $C = \{C_1, \dots, C_n\}$; locations
        $\{z_a \in \mathbb{R}^M : \|z_a\|_2 = 1\}$; masses $\{m_a \geqslant 0\}$;
        stabilizer $\zeta > 0$.
**Output:** clusters $\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_m\}$.
Compute pairwise distances $d_{ab} = \|z_a - z_b\|_2$ for all $a < b$;
Set shielding length $\lambda \leftarrow \text{median}\big(\{d_{ab} : a < b\}\big)$;
**foreach** pairs $(a, b)$ with $a < b$ **do**
    $A_{ab} \leftarrow \dfrac{m_a m_b}{d_{ab} + \zeta} \exp\Big(-\big(\tfrac{d_{ab}}{\lambda}\big)^2\Big)$;
    $S_a \leftarrow \sum_{b \neq a} A_{ab}$;
    $\widetilde{A}_{ab} \leftarrow A_{ab} / \sqrt{S_a S_b}$;
**end foreach**
Calculate Otsu[42] thresholding $\tau$;
**foreach** pairs $(a, b)$ with $a < b$ **do**
    **if** $\widetilde{A}_{ab} \geqslant \tau$ **then**
        union$(a, b)$;
    **end if**
**end foreach**
**return** $\mathcal{G}$;
```

- **Non-IID with Equal Size (Non-IID-ES).** Each client only has a limited subset of labels. However, the data volume per client is equal.
- **Non-IID.** Both label and data volume are different across clients. This dataset reflects more realistic conditions.

4.2 Experiment Setting

Experiments are conducted on 4 NVIDIA GeForce RTX 4090 GPUs. Local training on each client is 5 epochs per communication round, and use SGD with an initial learning rate of 0.01, decaying by a factor of 0.5 every 20 rounds. The 5 random seeds are used for the experiments, and the average results is the average with standard deviations. For MNIST data, GCFL use a CNN with two convolutional layers (32 and 64 filters) followed by a fully connected layer. For CIFAR-10 data, GCFL use a CNN includes two convolutional layers (64 and 128 filters) and two fully connected layers with 256 and 128 units.

4.3 Accuracy Results

To comprehensively evaluate model accuracy under different federation scales, this experiment consider three training settings:

- **Small-scale:** 50 total clients participate in FL, with full participation in each round.
- **Medium-scale:** 200 total clients participate in FL, with 25% of clients sampled per round.
- **Large-scale:** 500 total clients participate in FL, with 10% of clients sampled per round.

The Baseline model include non-clustering model (FedAvg[43], FedProx[44]), post-training clustering (CFL[9], FL+HC[19]), and in-training clustering (IFCA[9], AutoCFL[45], MTCFL[18], FedSoft[16], FedEM[17], CFL-GP[32], FedCCFA[34]).

When finish the training, each client evaluates the model it receives (global or cluster-personalized) on their own dataset; the framework's accuracy is computed as the average across all clients. Table 2 reports the test accuracy under the small-scale setting. Overall, GCFL achieve the best result in all NON-IID settings.

On the most challenging setting CIFAR-10/Non-IID, GCFL achieves 71.54%, which improves over the second-best model FL+HC by 1.18%. This result shows that, even under the extreme heterogeneity data environment, GCFL can also train several personalized models to effectively suppress the Non-IID problem.

Compared with the SOTA in-training clustering baseline FedCCFA on CIFAR-10, GCFL also improves 1.13% under Non-IID-ES, and 2.66% under Non-IID. This accuracy gain grows with heterogeneity, which shows that GCFL has better performance in more heterogeneous environments. This is because GCFL's gravitational clustering performs well, so clients with similar data are grouped into clusters with smaller intra-cluster distances and reduced inter-cluster noise. Therefore, it enhances personalization and achieves stronger models under Non-IID conditions.

It is also worth noting that under the IID setting, data are randomly partitioned across clients. The local objectives are nearly equal to the global objective. Therefore, in GCFL, clients naturally group into one or a few near-identical clusters, which limits the potential benefit of personalization. In this case, GCFL effectively behaves as a single global model, which explains why its accuracy is similar to the other baselines.

Also, the result shows that Post-training clustering typically outperforms in-training clustering. This is because post-training clustering methods group client after the initial FL run, and client models are closer to stationarity with less noise, thereby avoiding identity oscillations. However, the cost for this is the extra training time. Even under these favorable conditions, GCFL still achieves the highest accuracy in most cases, demonstrating a good balance between accuracy and complexity.

Table 2 Model accuracy of each method for Small-scale setting

Category	Method	CIFAR-10(Acc %)			MNIST(Acc %)		
		IID	Non-IID-ES	Non-IID	IID	Non-IID-ES	Non-IID
No Clustering	*FedAvg*	71.23 ± 0.25	59.21 ± 0.51	52.54 ± 1.68	96.31 ± 0.26	91.88 ± 0.44	90.29 ± 1.33
	FedProx	74.12 ± 0.22	58.42 ± 0.48	53.33 ± 1.55	96.57 ± 0.22	91.71 ± 0.45	90.76 ± 0.97
Post-training Clustering	*CFL*	77.64 ± 0.28	70.65 ± 0.55	68.10 ± 1.22	98.44 ± 0.28	95.70 ± 0.38	95.07 ± 0.95
	FL+HC	**78.21 ± 0.35**	73.58 ± 1.40	70.36 ± 1.85	98.59 ± 0.34	96.11 ± 0.45	95.82 ± 1.13
In-training Clustering	*IFCA*	76.33 ± 0.32	68.02 ± 0.60	62.25 ± 1.20	97.21 ± 0.18	95.18 ± 0.32	94.18 ± 0.69
	AutoCFL	75.45 ± 0.26	67.23 ± 0.56	61.39 ± 1.52	97.11 ± 0.23	94.81 ± 0.31	94.21 ± 0.61
	MTCFL	77.21 ± 0.24	68.54 ± 0.91	63.32 ± 1.94	98.07 ± 0.21	95.09 ± 0.77	94.41 ± 1.09
	FedSoft	77.19 ± 0.24	68.68 ± 0.52	65.33 ± 1.60	97.90 ± 0.25	95.10 ± 0.33	95.01 ± 0.66
	FedEM	76.70 ± 0.23	70.43 ± 0.45	67.44 ± 1.41	97.07 ± 0.24	96.20 ± 0.30	95.31 ± 0.58
	CFL-GP	78.05 ± 0.24	71.47 ± 0.43	68.34 ± 1.26	98.74 ± 0.17	96.55 ± 0.25	95.60 ± 0.50
	FedCCFA	78.13 ± 0.26	72.52 ± 0.41	68.88 ± 1.22	**98.78 ± 0.16**	96.48 ± 0.24	95.56 ± 0.47
Ours	*GCFL*	77.55 ± 0.22	**73.65 ± 0.40**	**71.54 ± 1.55**	98.65 ± 0.20	**96.69 ± 0.26**	**96.02 ± 0.55**

Table 3 Model accuracy of each method for Medium and Large scale settings.

Category	Method	CIFAR-10 (Acc %)						MNIST (Acc %)					
		Medium-scale			Large-scale			Medium-scale			Large-scale		
		IID	Non-IID-ES	Non-IID	IID	Non-IID-ES	Non-IID	IID	Non-IID-ES	Non-IID	IID	Non-IID-ES	Non-IID
No Clustering	*FedAvg*	68.21 ± 0.44	53.94 ± 0.97	44.63 ± 2.86	65.47 ± 0.56	49.88 ± 1.12	40.52 ± 3.21	95.42 ± 0.36	89.97 ± 0.74	86.41 ± 2.12	94.86 ± 0.41	88.91 ± 0.83	84.92 ± 2.36
	FedProx	70.63 ± 0.41	53.42 ± 0.94	45.12 ± 2.63	67.88 ± 0.53	49.41 ± 1.08	41.06 ± 2.98	95.61 ± 0.33	89.83 ± 0.76	86.92 ± 1.86	95.03 ± 0.39	88.74 ± 0.85	85.34 ± 2.07
Post-training Clustering	*CFL*	74.98 ± 0.49	68.94 ± 1.06	64.82 ± 2.22	73.92 ± 0.58	68.42 ± 1.18	63.71 ± 2.44	97.93 ± 0.26	94.88 ± 0.61	93.41 ± 1.22	97.42 ± 0.30	93.94 ± 0.68	92.28 ± 1.31
	FL+HC	**76.08 ± 0.51**	71.02 ± 1.36	66.71 ± 2.38	**74.65 ± 0.60**	**69.84 ± 1.47**	64.61 ± 2.57	98.11 ± 0.23	95.54 ± 0.65	94.68 ± 1.24	97.63 ± 0.28	**94.92 ± 0.71**	93.55 ± 1.33
In-training Clustering	*IFCA*	72.98 ± 0.58	62.91 ± 1.26	54.21 ± 2.97	69.82 ± 0.69	58.04 ± 1.44	48.92 ± 3.34	96.32 ± 0.35	92.88 ± 0.69	90.74 ± 1.41	95.44 ± 0.40	91.72 ± 0.80	88.96 ± 1.60
	AutoCFL	72.44 ± 0.56	62.27 ± 1.22	53.66 ± 3.06	69.21 ± 0.67	57.36 ± 1.41	48.31 ± 3.45	96.21 ± 0.36	92.64 ± 0.71	90.81 ± 1.38	95.31 ± 0.41	91.53 ± 0.82	89.02 ± 1.58
	MTCFL	73.61 ± 0.53	63.44 ± 1.41	54.92 ± 3.18	70.44 ± 0.64	58.73 ± 1.60	49.63 ± 3.56	97.28 ± 0.28	92.93 ± 1.05	91.11 ± 1.63	96.46 ± 0.33	91.84 ± 1.18	89.22 ± 1.84
	FedSoft	73.74 ± 0.54	63.88 ± 1.13	57.12 ± 2.76	70.61 ± 0.65	59.33 ± 1.31	51.86 ± 3.09	97.11 ± 0.30	93.07 ± 0.73	91.92 ± 1.46	96.32 ± 0.35	92.11 ± 0.85	90.03 ± 1.66
	FedEM	73.48 ± 0.55	66.21 ± 1.02	61.72 ± 2.59	70.38 ± 0.66	63.02 ± 1.21	57.94 ± 2.91	96.02 ± 0.38	94.41 ± 0.62	93.86 ± 1.22	95.21 ± 0.43	94.55 ± 0.70	93.10 ± 1.36
	CFL-GP	75.79 ± 0.52	69.62 ± 0.97	65.22 ± 2.28	73.96 ± 0.62	68.88 ± 1.14	62.84 ± 2.55	98.32 ± 0.22	95.62 ± 0.47	94.40 ± 1.10	**97.98 ± 0.27**	94.32 ± 0.56	92.73 ± 1.25
	FedCCFA	75.61 ± 0.53	70.68 ± 0.96	66.93 ± 2.24	74.08 ± 0.63	69.05 ± 1.13	64.12 ± 2.52	**98.36 ± 0.21**	95.49 ± 0.49	94.33 ± 1.09	97.82 ± 0.26	94.28 ± 0.57	92.64 ± 1.26
Ours	*GCFL*	75.73 ± 0.50	**71.36 ± 0.94**	**67.41 ± 2.31**	74.22 ± 0.60	69.31 ± 1.10	**64.82 ± 2.61**	98.29 ± 0.21	**95.86 ± 0.43**	**94.91 ± 1.04**	97.86 ± 0.26	94.46 ± 0.54	**93.72 ± 1.20**

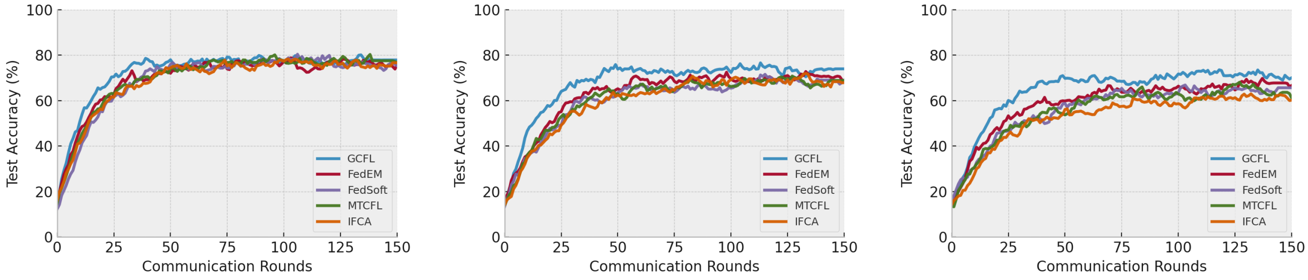
To test the scalability of GCFL, results with the medium and large scale setting under partial participation are reported in Table 3. On CIFAR-10 with non-IID data, FedAvg drops by 12.02%, indicating that partial participation can further exacerbate the non-IID issue. GCFL drops by only 6.72% on CIFAR-10 and 2.30% on MNIST, while still achieving the best performance under non-IID distributions. These results suggest that client clustering reduces cross-client interference by promoting more homogeneous updates within clusters, thereby improving robustness and scalability even under high client heterogeneity.

Moreover, as the federated learning setting scales up with partial participation, training becomes less stable, and this instability is exacerbated under non-IID data distributions. For instance, FedAvg exhibits a standard deviation of 3.21%, whereas GCFL achieves a lower standard deviation of 2.61%. This is because clustering reduces update conflicts and sampling-induced noise under non-IID data and partial participation.

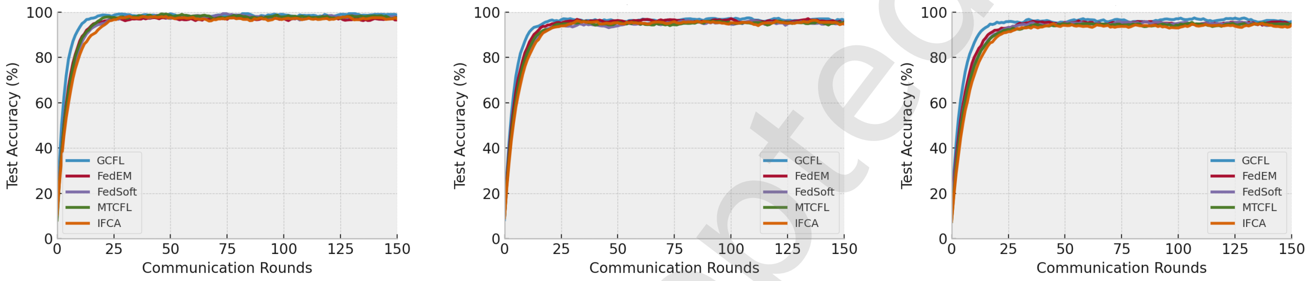
4.3.1 *Convergence Speed*

Then convergence speed is compared between GCFL and baselines. To ensure a fair comparison, all pre-training setting is removed (GCFL start clustering from round 0) and post-training methods are excluded because their additional retraining stage changes the round/time budget. As shown in Figure 2, accuracy is plotted against communication rounds under IID, Non-IID-ES, and Non-IID for CIFAR-10 and MNIST. GCFL's accuracy rises more quickly in the early rounds, and show fewer oscillations. This advantage is most evident in Non-IID data conditions, where baselines improve more slowly and GCFL stabilizes earlier.

To quantify this convergence speed, Experiment also design two metrics to evaluate. $T_{0.95}$ is the number of rounds each model required to reach 95% of their own final accuracy; and AUC_{50} is the mean accuracy over the first 50 rounds, where $AUC_{50} = \frac{1}{50}\sum_{t=1}^{50} Acc_t$.



(a) CIFAR-10 (IID) (b) CIFAR-10 (Non-IID-ES) (c) CIFAR-10 (Non-IID)



(d) MNIST (IID) (e) MNIST (Non-IID-ES) (f) MNIST (Non-IID)

Fig. 2 Accuracy–round curves comparing GCFL and baselines on CIFAR-10 and MNIST

Table 4 Convergence Speed for In-training clusters

Method	CIFAR-10 (Non-IID-ES)		CIFAR-10 (Non-IID)		MNIST (Non-IID-ES)		MNIST (Non-IID)	
	$T_{0.95}\downarrow$	$AUC_{50}\uparrow$	$T_{0.95}\downarrow$	$AUC_{50}\uparrow$	$T_{0.95}\downarrow$	$AUC_{50}\uparrow$	$T_{0.95}\downarrow$	$AUC_{50}\uparrow$
GCFL	**44**	**41.31**	**52**	**37.15**	**12**	**79.25**	**14**	**76.20**
FedEM	58	34.11	68	28.62	15	75.53	18	72.12
FedSoft	63	31.12	74	26.65	17	73.43	20	69.84
MTCFL	60	31.43	71	27.51	16	74.11	19	70.55
IFCA	68	29.89	82	25.41	18	71.32	22	68.73

Table 4 shows that GCFL achieves the fastest convergence in all data settings. On CIFAR-10, GCFL reaches a stable accuracy roughly 10 rounds earlier than the second best methods; on MNIST, where all methods converge quickly, GCFL still maintains higher early round accuracy. These results double check the curve-level observations, confirming that GCFL converges faster and have better training accuracy.

4.4 Ablation Studies

To verify the function of GCFL component, the ablation studies is conducted. Specifically, GCFL ablate five core components as follow:

- Massless: Set all client masses to 1, remove mass modeling as a whole without tuning its sub-terms;
- Param-space clusteringL: Remove the probe set in server, and calculate the distance by there parameter-space distances;
- w/o shielding: Remove gravitational shielding;
- w/o symmetric normalization: Remove symmetric normalization;
- No δ_{init}: Start clustering from round 0.

Table 5 shows the results, which have two metrics including model accuracy and average number of clusters. The full GCFL gets the highest accuracy across all datasets, which indicates that each component is indispensable in this framework. The most essential part is gravitational shielding. Removing it produces the largest degradation. On CIFAR-10, the accuracy drops by 10.49% and 9.73% and the average cluster number decreases to 2.1 under Non-IID-ES and 2.4 under Non-IID. This result indicates that, clients tend to merge into a single cluster, thereby losing personalization. Massless

Table 5 Ablation of GCFL Components on CIFAR-10 and MNIST

Variant	CIFAR-10 (Non-IID-ES)		CIFAR-10 (Non-IID)		MNIST (Non-IID-ES)		MNIST (Non-IID)	
	Acc %	Avg. m	Acc %	Avg. m	Acc %	Avg. m	Acc %	Avg. m
GCFL (full)	73.65 ± 0.40	4.3	71.54 ± 1.55	5.1	96.69 ± 0.26	3.9	96.02 ± 0.55	4.5
Massless	68.31 ± 0.65	3.6	64.91 ± 1.85	3.7	95.32 ± 0.40	3.1	94.12 ± 0.60	3.3
Param-space	70.31 ± 0.62	4.5	68.12 ± 1.78	6.2	96.21 ± 0.42	4.1	95.54 ± 0.58	4.7
w/o shielding	63.16 ± 0.70	2.1	61.81 ± 1.90	2.4	93.42 ± 0.45	1.8	94.21 ± 0.62	2.0
w/o symmetric normalization	72.05 ± 0.85	3.5	69.70 ± 1.95	3.9	95.95 ± 0.48	3.4	95.10 ± 0.65	3.9
No δ_{init}	71.61 ± 0.75	4.4	69.15 ± 1.95	5.3	96.31 ± 0.38	4.1	95.84 ± 0.60	4.5

also decreases accuracy and results in fewer clusters. These results indicate that the ”mass” accurately represents client importance and reduces intra-cluster distance. Also, using raw parameters to compute distances gives worse results, which confirms that a small, label-balanced probe set is more accurate at localizing models. Starting clustering from round 0 will make a small decrease in accuracy; the overall effect is not large, but deferring clustering until a initial global model remains preferable.

5 Conclusions

This manuscript presents GCFL, an adaptive in-training federated clustering framework. GCFL treats clients as bodies in a latent space, whose locations are is their local models and masses represent the client importance. GCFL employs a gravitational clustering algorithm that forms stable clusters without predefining a large number of sensitive hyperparameters, thereby achieving higher accuracy and faster convergence under Non-IID conditions.

The future work will strengthen the gravitational clustering by moving from a static attraction matrix to a force-driven, dynamic update of client locations: in each round, the server will integrate the resultant forces with velocity to update each client’s position, allowing clients to drift smoothly toward nearby groups and further improving stability and convergence speed.

Acknowledgment

This research work was partly supported by National Natural Science Foundation of China under grants No. 62271125 and the Sichuan Provincial Department of Science and Technology, Natural Science Fund for Innovative Research Group Project NO.2024NSFTD0033.

Reference

[1] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, Federated learning in mobile edge networks: A comprehensive survey, *IEEE communications surveys & tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[2] C. Li, B. Gu, Z. Zhao, Y. Qu, G. Xin, J. Huo, and L. Gao, Federated transfer learning for on-device llms efficient fine tuning optimization, *Big Data Mining and Analytics*, vol. 8, no. 2, pp. 430–446, 2025.

[3] R. S. Antunes, C. André da Costa, A. Küderle, I. A. Yari, and B. Eskofier, Federated learning for healthcare: Systematic review and architecture proposal, *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–23, 2022.

[4] A. Al-Dailami, H. Kuang, and J. Wang, Fedcomdist: Towards effective personalized federated learning for patient outcome prediction using multi-center electronic medical records, *IEEE Journal of Biomedical and Health Informatics*, 2025.

[5] D. Byrd and A. Polychroniadou, Differentially private secure multi-party computation for federated learning in financial applications, in *Proceedings of the first ACM international conference on AI in finance*, 2020, pp. 1–9.

[6] M. Yao, S. Qi, Z. Tian, Q. Li, Y. Han, H. Li, and Y. Qi, Quantifying bytes: Understanding practical value of data assets in federated learning, *Tsinghua Science and Technology*, vol. 30, no. 1, pp. 135–147, 2024.

[7] K. Chen, W. Li, J. Cao, B. Mi, and J. Shen, Optimizing federated incremental learning: Efficient malicious data removal for big data analytics, *Tsinghua Science and Technology*, 2025.

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

[9] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, An efficient framework for clustered federated learning, *Advances in neural information processing systems*, vol. 33, pp. 19 586–19 597, 2020.

[10] L. Song, J. Li, H. Jiang, S. Wei, and Y. Guo, Chpfl: Clustered adaptive hierarchical federated learning for edge-level personalization, *High-Confidence Computing*, p. 100343, 2025.

[11] Q. Zia, S. Zhu, H. Wang, Z. Iqbal, and Y. Li, Hierarchical federated transfer learning in digital twin-based vehicular networks, *High-Confidence Computing*, p. 100303, 2025.

[12] T. Liang, C. Yuan, C. Lu, Y. Li, J. Yuan, and Y. Yin, Effi-

cient one-off clustering for personalized federated learning, *Knowledge-Based Systems*, vol. 277, p. 110813, 2023.

[13] E. Yu, Z. Ye, Z. Zhang, L. Qian, and M. Xie, A federated recommendation algorithm based on user clustering and meta-learning, *Applied Soft Computing*, vol. 158, p. 111483, 2024.

[14] S. Vahidian, M. Morafah, W. Wang, V. Kungurtsev, C. Chen, M. Shah, and B. Lin, Efficient distribution similarity identification in clustered federated learning via principal angles between client data subspaces, in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 8, 2023, pp. 10 043–10 052.

[15] G. Lu, K. Li, X. Wang, Z. Liu, Z. Cai, and W. Li, Neural-based inexact graph de-anonymization, *High-Confidence Computing*, vol. 4, no. 1, p. 100186, 2024.

[16] Y. Ruan and C. Joe-Wong, Fedsoft: Soft clustered federated learning with proximal local updating, in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 7, 2022, pp. 8124–8131.

[17] O. Marfoq, G. Neglia, A. Bellet, L. Kameni, and R. Vidal, Federated multi-task learning under a mixture of distributions, *Advances in neural information processing systems*, vol. 34, pp. 15 434–15 447, 2021.

[18] F. Sattler, K.-R. Müller, and W. Samek, Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints, *IEEE transactions on neural networks and learning systems*, vol. 32, no. 8, pp. 3710–3722, 2020.

[19] C. Briggs, Z. Fan, and P. Andras, Federated learning with hierarchical clustering of local updates to improve training on non-iid data, in *2020 international joint conference on neural networks (IJCNN)*. IEEE, 2020, pp. 1–9.

[20] M. K. Zuziak, R. Pellungrini, and S. Rinzivillo, One-shot clustering for federated learning, in *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2024, pp. 8108–8117.

[21] D. Scheliga, P. Maeder, and M. Seeland, Feature-based dataset fingerprinting for clustered federated learning on medical image data, *Applied Artificial Intelligence*, vol. 38, no. 1, p. 2394756, 2024.

[22] A. Elhussein and G. Gürsoy, Privacy-preserving patient clustering for personalized federated learnings, in *Machine Learning for Healthcare Conference*. PMLR, 2023, pp. 150–166.

[23] D. K. Dennis, T. Li, and V. Smith, Heterogeneity for the win: One-shot federated clustering, in *International conference on machine learning*. PMLR, 2021, pp. 2611–2620.

[24] Y. Wang, W. Pang, D. Wang, and W. Pedrycz, One-shot federated k-means clustering based on density cores, *Authorea Preprints*, 2023.

[25] M. Li, D. Avdiukhin, R. Shahout, N. Ivkin, V. Braverman, and M. Yu, Federated learning clients clustering with adaptation to data drifts, *arXiv preprint arXiv:2411.01580*, 2024.

[26] M. A. Helcig and S. Nastic, Fedccl: Federated clustered continual learning framework for privacy-focused energy forecasting, in *2025 IEEE 9th International Conference on Fog and Edge Computing (ICFEC)*. IEEE, 2025, pp. 50–57.

[27] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, A density-based algorithm for discovering clusters in large spatial databases with noise, in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[28] G. Long, M. Xie, T. Shen, T. Zhou, X. Wang, and J. Jiang, Multi-center federated learning: clients clustering for better personalization, *World Wide Web*, vol. 26, no. 1, pp. 481–500, 2023.

[29] R. Du, S. Xu, R. Zhang, L. Xu, and H. Xia, A dynamic adaptive iterative clustered federated learning scheme, *Knowledge-Based Systems*, vol. 276, p. 110741, 2023.

[30] B. Liu, Y. Ma, Z. Zhou, Y. Shi, S. Li, and Y. Tong, Casa: Clustered federated learning with asynchronous clients, in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 1851–1862.

[31] I. Lin, O. Yagan, C. Joe-Wong *et al.*, Fedspd: A soft-clustering approach for personalized decentralized federated learning, *arXiv preprint arXiv:2410.18862*, 2024.

[32] H. Kim, H. Kim, and G. De Veciana, Clustered federated learning via gradient-based partitioning, in *Forty-first international conference on machine learning*, 2024.

[33] E. Gu, Y. Chen, H. Wen, X. Cai, and D. Han, Novel clustered federated learning based on local loss, *arXiv preprint arXiv:2407.09360*, 2024.

[34] J. Chen, J. Xue, Y. Wang, Z. Liu, and L. Huang, Classifier clustering and feature alignment for federated learning under distributed concept drift, *Advances in Neural Information Processing Systems*, vol. 37, pp. 81 360–81 388, 2024.

[35] M. Zhang, T. Zhang, Y. Cheng, C. Bao, H. Cao, D. Jiang, and L. Xu, Communication-efficient clustered federated learning via model distance, *Machine learning*, vol. 113, no. 6, pp. 3869–3888, 2024.

[36] H. Vardhan, A. Ghosh, and A. Mazumdar, An improved federated clustering algorithm with model-based clustering, *Transactions on machine learning research*, 2024.

[37] X. Xue, H. Mao, and Q. Li, Dag-acfl: Asynchronous clustered federated learning based on dag-dlt, *arXiv preprint arXiv:2308.13158*, 2023.

[38] Y. Guo, X. Tang, and T. Lin, Enhancing clustered federated learning: Integration of strategies and improved methodologies, in *The Thirteenth International Conference on Learning Representations*.

[39] Z. Liu, Z. Shen, P. Zhou, Q. Zheng, and J. Jin, Fedhc: A hierarchical clustered federated learning framework for satellite networks, *arXiv preprint arXiv:2502.12783*, 2025.

[40] D. Jhunjhunwala, S. Wang, and G. Joshi, Fedfisher: Leveraging fisher information for one-shot federated learning, in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2024, pp. 1612–1620.

[41] J. Zhang and Z. Qiao, Sofl: Clustered federated learning based on dual clustering for heterogeneous data, *Electronics*, vol. 13, no. 18, p. 3682, 2024.

[42] D. Liu and J. Yu, Otsu method and k-means, in *2009 Ninth International conference on hybrid intelligent systems*, vol. 1. IEEE, 2009, pp. 344–349.

[43] 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.

[44] 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.

[45] B. Gong, T. Xing, Z. Liu, W. Xi, and X. Chen, Adaptive client clustering for efficient federated learning over non-iid and imbalanced data, *IEEE Transactions on Big Data*, vol. 10, no. 6, pp. 1051–1065, 2022.

Author biography

Guangxi Lu Guangxi Lu received his Ph.D. degree in Computer Science from Georgia State University in 2023. He is currently a Postdoctoral Researcher with the School of Computer Science and Engineering at the University of Electronic Science and Technology of China. His main research interest is in data security and privacy protection, emphasizing the importance of security and privacy in decentralized computing environments.

Lizong Zhang Lizong Zhang received the BSc degree from University of Electronic Science and Technology of China in 2003, the MSc and PhD degrees in computer science from Staffordshire University, UK in 2007 and 2013, respectively. He is a full professor in computing science at the School of Computer Science and Engineering, University of Electronic Sciences and Technology of China, China. His current research interests include AI, machine learning, knowledge management, and computer vision.

Chong Mu Chong Mu received his B.E. degree in School of Software from Hefei University of Technology (HFUT) in 2020. He obtained his Ph.D. degree in the School of Information and Software Engineering, University of Electronic Science and Technology of China (UESTC) in 2024. He is currently a postdoctoral researcher at University of Electronic Science and Technology of China (UESTC). His research interests include knowledge graph completion and inductive reasoning.

Haoji Zhang Haoji Zhang received the B.Eng. in Telecommunications Engineering and Management from Beijing University of Posts and Telecommunications in 2023 and the M.Sc. in Computer Control and Automation from Nanyang Technological University, Singapore, in 2025. He is currently a research assistant with the School of Computer Science and Engineering at the University of Electronic Science and Technology of China. His research focuses on robotic manipulation and grasping, multimodal fusion, object detection, and computer vision.