Clustered Federated Learning via Gradient-based Partitioning

Heasung Kim [1] **Hyeji Kim** [1] **Gustavo de Veciana** [1]

Abstract

Clustered Federated Learning (CFL) is a promising distributed learning framework that addresses data heterogeneity issues across multiple clients by grouping clients and providing a shared generalized model for each group. However, under privacy-preserving federated learning protocols where there is no direct sharing of clients' local datasets, existing approaches often fail to find optimal client groupings resulting in sub-optimal performance. In this paper, we propose a novel CFL algorithm that achieves robust clustering and learning performance. Conceptually, our algorithm groups clients that exhibit similarity in their model updates by periodically accumulating and clustering the gradients that clients compute for various models. The proposed algorithm is shown to achieve a near-optimal error rate for stochastic convergence to optimal models under mild conditions. We present a detailed analysis of the algorithm along with an evaluation on several CFL benchmarks demonstrating that it outperforms existing approaches in terms of convergence speed, clustering accuracy, and task performance.

1. Introduction

The exploitation of distributed data via privacy preserving cooperative learning algorithms is one of the foundational challenges of modern machine learning. In this regard, substantial attention has been given to *Federated Learning* (FL), a distributed learning framework that provides a degree of data privacy by only sharing information about locally trained models versus raw data (McMahan et al., 2017). Indeed FL has proved to be widely successful for training one global model for multiple local machines (Bonawitz et al., 2019; Yang et al., 2020; Niknam et al., 2020; Rieke et al., 2020).

When dealing with heterogeneous local datasets, the training of a shared global model for all clients may not yield optimal results. To address this issue, researchers have proposed model personalization techniques (Kulkarni et al., 2020) that involve clients collaboratively learning a global model and then fine-tuning it to each client's data (Fallah et al., 2020; Singhal et al., 2021; Smith et al., 2017; Wang et al., 2019; Yu et al., 2020). However, in real-world scenarios, the effectiveness of fine-tuning may be limited by the small size of clients' local datasets. Moreover, the use of a shared large backbone model across clients may be impractical in scenarios where one wishes to obtain lightweight models, e.g., for IoT devices, or when clients lack common features.

The Clustered Federated Learning (CFL) framework (Sattler et al., 2020; Mansour et al., 2020; Ghosh et al., 2020) can effectively address these issues by clustering/grouping clients to share a few models, thus creating models tailored for each cluster, rather than one personalized to each client. CFL promotes cooperation among clients with shared local distributions only rather than aggregating all the clients, resulting in more accurate lightweight models.

While CFL has the potential to deliver tremendous advantages, algorithmic developments have proved to be remarkably challenging due to the need to jointly cluster and train over distributed data. Recent theoretically grounded CFL algorithms group clients based on the models which currently provide each the best performance (Ghosh et al., 2020; Mansour et al., 2020), or recursively perform bipartitioning when clients' gradients differ on a converged global model (Sattler et al., 2020). Unfortunately, the state-of-the-art CFL algorithms face difficulties in various relevant settings including e.g., linear regression tasks (See 5.1). This raises the following question:

"What are statistically well-founded and empirically reliable features that could be used for clustering clients during FL training with limited access to raw datasets and how should clustering be performed?"

In this paper, we provide an answer to this question and propose a new class of robust CFL algorithms that can handle the abovementioned challenges. The *main contributions* of

[1]Department of Electrical and Computer Engineering, The University of Texas at Austin, TX, USA. Correspondence to: Heasung Kim <heasung.kim@utexas.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Source code: https://github.com/Heasung-Kim/clustered-federated-learning-via-gradient-based-partitioning

this paper can be summarized as follows.

Algorithm design: We devise a novel CFL algorithm motivated by the intuition that clients with similar distributions would share similar gradients and thus good prospects for clustering. The proposed algorithm periodically computes gradients based on clients' local datasets over a set of models, accumulates this information, and partitions clients into groups that exhibit similarity in their accumulated gradients via spectral clustering. The clients are assigned to models based on the estimated client clusters, and the models are updated accordingly. Given the above design principle, in the sequel, we refer to our algorithm as *CFL-GP* (Clustered Federated Learning via Gradient-based Partitioning).

Theoretical analysis: To the best of our knowledge, CFL-GP is the first CFL algorithm with provable guarantees for convergence to both an optimal client clustering and set of models without requiring assumptions on the algorithm's initial conditions on the model parameters. This is achieved by combining recent ideas from spectral clustering and leveraging the Gaussianity of noise in high dimensional model gradient vectors. This distinguishes CFL-GP from other state-of-the-art approaches which either require a sufficiently large gap between optimal models associated with client clusters and/or require good initial conditions.

Evaluation: We demonstrate our approach's real-world effectiveness via extensive experiments. These include synthetic setups wherein we can explore CFL-GP's robustness to various parameters/settings and large-scale industrial applications. We find that for many representative CFL benchmark tasks, CFL-GP achieves optimal clustering significantly faster, up to 100 times faster than existing state-of-the-art algorithms that frequently struggle to achieve accurate clustering results. As a result we find that CFL-GP achieves better performance at lower overall communication and computational costs. Our comprehensive evaluation also includes an exploration of the impact of the model gap, batch size, number of clients, mixture of distributions, and gradient vector compression.

2. System Model

We consider a federated learning system with C clients, where the c-th client is modelled as having a minibatch dataset generated from one of D distinct distributions, i.e., $\mathcal{D}(c) \in \{\mathcal{D}_1, \cdots, \mathcal{D}_D\}$, where $D \leq C$. This results in a natural clustering of the clients defined as $\mathcal{S}_i^* = \{c \in [C] : \mathcal{D}(c) = \mathcal{D}_i\}$ for $i \in [D]$. Here, $[C]$ denotes a set of consecutive natural numbers $\{1, 2, ...C\}, C \in \mathbb{N}$.

Let $F(\mathcal{D}, \boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[l(\mathbf{x}, \boldsymbol{\theta})]$ denote the population loss, i.e., expectated loss l of a data sample $\mathbf{x}$ generated from the distribution $\mathcal{D}$ on a model $\boldsymbol{\theta} \in \Theta$. Here $\Theta \subset \mathbb{R}^d$ denotes the space of parameterized models. The following problem definition formalizes CFL.

Problem 1 (Clustered Federated Learning)**.** *Given a pre-specified number of clusters K where $K \leq C$, we aim to find the optimal clustering $\pi : [C] \to [K]$ and K models $\{\boldsymbol{\theta}_k\}_{k=1}^K$ which minimizes the sum loss, i.e.,*

$$\min_{\{\boldsymbol{\theta}_k\}_{k=1}^K, \pi} \quad \Big\{ \sum_{c=1}^{C} F(\mathcal{D}(c), \boldsymbol{\theta}_{\pi(c)}) \Big| \pi \in \Pi \Big\} \tag{1}$$

where $\Pi = \{\pi : [C] \xrightarrow{\mathsf{S}} [K]\}$ denotes a surjective function space, indicating that the problem involves clustering.

Let $\mathcal{S}_k = \{c \in [C] : \pi(c) = k\}$ denote the set of clients assigned to the k-th model for $k \in [K]$, i.e., clusters arising from π. Then, the objective function in (1) can be equivalently expressed as $\sum_{k=1}^K \sum_{c\in\mathcal{S}_k} F(\mathcal{D}(c), \boldsymbol{\theta}_k)$.

During the training, each client c is modelled as capable of sampling a finite minibatch $\mathcal{X}_c$. We define the empirical loss function $f(\mathcal{X}_c, \boldsymbol{\theta})$ and its corresponding gradient $\nabla f(\mathcal{X}_c, \boldsymbol{\theta})$ associated with $\mathcal{X}_c$ as

$$f(\mathcal{X}_c, \boldsymbol{\theta}) = \frac{1}{|\mathcal{X}_c|} \sum_{\mathbf{x}\in\mathcal{X}_c} l(\mathbf{x}, \boldsymbol{\theta}) \tag{2}$$

$$\nabla f(\mathcal{X}_c, \boldsymbol{\theta}) = \nabla F(\mathcal{D}(c), \boldsymbol{\theta}) + \mathbf{e}_c(\boldsymbol{\theta}) \tag{3}$$

where $\nabla F(\mathcal{D}(c), \boldsymbol{\theta})$ denotes the gradient of the population loss with respect to the model parameters $\boldsymbol{\theta}$, and $\mathbf{e}_c(\boldsymbol{\theta})$ corresponds to a zero-mean Stochastic Gradient Noise (SGN) vector modeling noise associated with minibatch sampling, following the convention of the stochastic gradient descent analysis (Ahn et al., 2012; Chen et al., 2014; Zhu et al., 2019).

Notation: We use superscripts within parentheses, e.g., $\boldsymbol{\theta}^{(t)}$ to signify the t-th iteration. We write $x \lesssim y$ if there exists a universal constant C_u such that $x \leq C_u y$ where $C_u > 0$. We use $\mathcal{O}$ to denote the order of functions.

3. Algorithm

CFL-GP is based on the intuition that clients whose model gradient updates align during the learning process should be grouped into the same cluster. Due to potential noise and high dimensionality of the gradients, exploiting these as clustering features while maintaining theoretical optimality can be challenging. To tackle this issue, we propose a novel algorithm that enables us to effectively leverage these features without compromising optimality.

In CFL-GP, the Central Unit (CU) maintains and updates K models $\{\boldsymbol{\theta}_k\}_{k=1}^K$ and K clusters of clients $\{\mathcal{S}_k\}_{k=1}^K$, where each cluster denotes a subset of clients whose distributions are believed to be similar and thus assigned to a shared common model. As depicted in the Algorithm 1 panel the

Algorithm 1 CFL-GP

Input: K initial models $\{\boldsymbol{\theta}_k^{(0)}\}_{k=1}^K$, K initial clusters $\{\mathcal{S}_k^{(0)}\}_{k=1}^K$, clustering period P, learning rate $\gamma_k^{(t)}$, gradient features initialized as $\boldsymbol{g}_c^{(-1)} = \mathbf{0}\ \forall c \in [C]$, feature moving average factor $\{\beta_t\}_{t=1}^T$, and the number of cluster updates T_{cl}

Output: K trained models $\{\boldsymbol{\theta}_k^{(T)}\}_{k=1}^K$ and K updated clusters $\{\mathcal{S}_k^{(T)}\}_{k=1}^K$

1: **for** $t = 0$ **to** T **do**
2: **for** $k = 1$ **to** K in parallel **do** ▷ *Model Update*
3: CU transmits $\boldsymbol{\theta}_k^{(t)}$ to the clients in $\mathcal{S}_k^{(t)}$ and receives $\nabla f(\mathcal{X}_c^{(t)}, \boldsymbol{\theta}_k^{(t)})\ \forall c \in \mathcal{S}_k^{(t)}$
4: CU updates $\boldsymbol{\theta}_k^{(t+1)} := \boldsymbol{\theta}_k^{(t)} - \gamma_k^{(t)} \sum_{c \in \mathcal{S}_k^{(t)}} \nabla f(\mathcal{X}_c^{(t)}, \boldsymbol{\theta}_k^{(t)})$
5: **if** $t \bmod P = 1$ and $t < T_{\text{cl}}$ **then** ▷ *Cluster Update*
6: Select $\bar{k} \in [K]$ in a round-robin manner
7: CU broadcasts $\boldsymbol{\theta}_{\bar{k}}^{(t)}$ and receives $\nabla f(\mathcal{X}_c^{(t)}, \boldsymbol{\theta}_{\bar{k}}^{(t)})\ \forall c \in [C]$
8: CU updates $\boldsymbol{g}_{c,\bar{k}}^{(t)} := (1-\beta_t)\boldsymbol{g}_{c,\bar{k}}^{(t-1)} + \beta_t \nabla f(\mathcal{X}_c^{(t)}, \boldsymbol{\theta}_{\bar{k}}^{(t)})\ \forall c \in [C]$
9: $\boldsymbol{g}_{c,k'}^{(t)} := \boldsymbol{g}_{c,k'}^{(t-1)}\ \forall c \in [C], k' \in [K]$ s.t. $k' \neq \bar{k}$
10: $\{\mathcal{S}_k^{(t+1)}\}_{k=1}^K := \textsc{SpectralClustering}(\{\boldsymbol{g}_c^{(t)}\}_{c=1}^C, \{\mathcal{S}_k^{(t)}\}_{k=1}^K)$
11: **else**
12: $\mathcal{S}_k^{(t+1)} := \mathcal{S}_k^{(t)}\ \forall k \in [K]$ and $\boldsymbol{g}_c^{(t)} := \boldsymbol{g}_c^{(t-1)}\ \forall c \in [C]$

CU iteratively updates both the model parameters and the client clusters.

Model update (Lines 2-4). Each of the K models is updated based on the gradients of clients in the associated cluster. As seen in Lines 2-4, the CU sends the k-th model $\boldsymbol{\theta}_k^{(t)}$ to every client in the k-th cluster $\mathcal{S}_k^{(t)}$ and collects their gradients $\nabla f(\mathcal{X}_c^{(t)}, \boldsymbol{\theta}_k^{(t)})$. Then it updates the k-th model as $\boldsymbol{\theta}_k^{(t+1)} := \boldsymbol{\theta}_k^{(t)} - \gamma_k^{(t)} \sum_{c \in \mathcal{S}_k^{(t)}} \nabla f(\mathcal{X}_c^{(t)}, \boldsymbol{\theta}_k^{(t)})$ for all $k \in [K]$.

Cluster update (Lines 5-12). Every P iterations, the CU updates the clustering of clients $\{\mathcal{S}_k^{(t)}\}_{k=1}^K$ by collecting their gradients on various common models and performing spectral clustering, provided that t remains within the clustering iteration threshold T_{cl}. Specifically, as explained in the next paragraph, we construct a feature vector $\boldsymbol{g}_c^{(t)} \in \mathbb{R}^{Kd}$ (K d-dimensional vectors) for each client c by concatenating its averaged gradients associated with K different models. This permits the clustering of clients based on the similarity of their gradients across time and diverse models.

As shown in Lines 5-7, the gradient data is collected by periodically broadcasting (in a round-robin manner) *one* of the K models, denoted by $\bar{k}$, to all clients and collecting the associated gradients from each client. As shown in Line 8, in each clustering period, the feature vector is smoothly updated to incorporate the newly received gradient information. We denote the $\bar{k}$-th block of $\boldsymbol{g}_c^{(t)}$ as $\boldsymbol{g}_{c,\bar{k}}^{(t)}$, which is defined by the $(\bar{k}d-d+1:\bar{k}d)$-th elements of $\boldsymbol{g}_c^{(t)}$. The $\bar{k}$-th block of the feature vector is smoothly updated as $\boldsymbol{g}_{c,\bar{k}}^{(t)} := (1-\beta_t)\boldsymbol{g}_{c,\bar{k}}^{(t-1)} + \beta_t \nabla f(\mathcal{X}_c^{(t)}, \boldsymbol{\theta}_{\bar{k}}^{(t)})$, where β_t denotes the moving average factor. The remaining parts of the clients' feature vectors are maintained as in Line 9. We set $\beta_t = \frac{1}{\lfloor t/(KP) \rfloor + 1}$ which means that we take the cumulative averaging for the feature vectors.

As gradients are high-dimensional, clustering them directly may be infeasible. To address this challenge, in Line 10, we utilize spectral clustering on the clients' feature vectors to group them based on similarity. We demonstrate through theoretical analysis and experimentation that even when the dimensionality of the gradient information is significantly reduced, as elaborated in the next paragraph, our method can efficiently perform clustering without sacrificing optimality.

Spectral clustering. By *Spectral clustering* here we refer to a family of methods involving dimensionality reduction and clustering (Löffler et al., 2021). The subroutine SPECTRALCLUSTERING is performed in three steps; (a) First is the dimensionality reduction. Let $\mathbf{G}^{(t)} = (\boldsymbol{g}_1^{(t)} \cdots \boldsymbol{g}_C^{(t)}) \in \mathbb{R}^{Kd \times C}$ denote the matrix form of the clients feature vectors. The CU computes $\hat{\boldsymbol{U}}^{(t)} \in \mathbb{R}^{Kd \times K}$ whose k-th column is the singular vector of $\mathbf{G}^{(t)}$ corresponding to the k-th largest singular value. Let $\boldsymbol{g}_c'^{(t)} = \hat{\boldsymbol{U}}^{(t)\top} \boldsymbol{g}_c^{(t)} \in \mathbb{R}^K$ denote the c-th client's projected feature vector, achieving a d-fold dimensionality reduction from $\boldsymbol{g}_c^{(t)} \in \mathbb{R}^{Kd}$ to $\boldsymbol{g}_c'^{(t)} \in \mathbb{R}^K$. ($b$) Next, the CU performs K-means clustering to partition the clients into K groups. (c) Finally, we assign a unique model to each group, resulting in K labeled clusters $\{\mathcal{S}_k^{(t+1)}\}_{k=1}^K$. Specifically, among all possible assignments of models to groups, we choose one maximally aligned and coherent with the previous assignment, i.e., one that maximizes the number of clients whose new model assignment is equal to the previous one. A more formal description of Algorithm 1

is provided in Appendix B.

Next, we present several techniques to improve the efficiency, complexity, and scalability of CFL-GP for practical deployment.

Model averaging protocol. The gradient averaging-based model update process of Lines 3-4 in Algorithm 1 can be replaced by the model averaging method for communication efficiency, where the clients perform multi-step local updates for the model update. See Appendix C.

Complexity. The primary contributor to the computation complexity of CFL-GP is the computation of the leading singular vectors of the gradient profile matrix within the spectral clustering, which can be rapidly solved by recent leading singular vectors recovery techniques, e.g., (Shamir, 2015) through additional iterations where the runtime is logarithmic in the required accuracy (Shamir, 2016). The spectral clustering ultimately leads CFL-GP to achieve much lower communication and computation costs as it quickly achieves a correct clustering (Section 4). Empirical results show that CFL-GP can achieve *optimal clustering with only a few spectral clustering updates* (up to 100 times faster than existing methods in various benchmarks as shown in Section 5 and Appendix D).

Scalability. To improve the scalability of CFL-GP in handling large neural networks (e.g., $> 10^6$ parameters), one can let clients send *compressed* gradients for clustering purposes in Line 7. We observe that effective spectral clustering can still be performed when gradients are compressed by more than $100\times$, which is in line with recent findings that gradients or task objective landscapes have *low intrinsic dimensionality* in many tasks (Li et al., 2018; Aghajanyan et al., 2020; Hu et al., 2021a). See Appendix E for more details.

4. Theoretical Analysis

In this section, we show that under the assumption that the gradients' noise is Gaussian, CFL-GP eventually achieves proper client clustering and convergence to the optimal model parameters with high probability. For analysis purposes, we shall assume that the number of distinct distributions D clients can have is equal to the number of models K to be trained.

For a given distribution $\mathcal{D}_k$, we let $\boldsymbol{\theta}_k^*$ denote an optimal model, i.e., $\boldsymbol{\theta}_k^* = \operatorname{argmin}_{\boldsymbol{\theta}\in\Theta}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_k}[l(\mathbf{x},\boldsymbol{\theta})]$. In an ideal setting the clients in S_k^* sharing distribution $\mathcal{D}_k$ would eventually be grouped together and assigned to the associated model.

It is imperative to note that in Algorithm 1, the model $\boldsymbol{\theta}_k^{(t)}$ and client cluster $\mathcal{S}_k^{(t)}$ at iteration t are not tied to the k-th distribution. The index k in $\boldsymbol{\theta}_k^{(t)}$ only signifies its ordinal position among K models rather than a specific distribution association. The k-th model may engage with a variety of clients during training, each possibly having a different data distribution index. Given this variability, an appropriate notion for tracking the convergence of the models becomes necessary.

To address this challenge, we introduce a new notion in model indexing for clustering and convergence analysis: for a given distribution $\mathcal{D}_k$, we consider a model which the *plurality* of clients in S_k^* agree. This is formally defined as follows.

Definition 1 (Client plurality model)**.** The *client-plurality model* for clients with distribution $\mathcal{D}_k$, denoted $\hat{\boldsymbol{\theta}}_k^{(t)}$, is given as $\hat{\boldsymbol{\theta}}_k^{(t)} = \boldsymbol{\theta}_{M(k,t)}^{(t)}$ where $M(k,t) \in \arg\max_{m\in[K]} |\mathcal{S}_k^* \cap \mathcal{S}_m^{(t)}|$ and with ties are broken arbitrarily. We shall see plurality will more often than not be in fact the majority as clients become properly grouped.

In other words, this formulation provides an index of a model predominantly chosen by clients whose underlying true data distribution is $\mathcal{D}_k$ at iteration t.

Our convergence analysis regarding client clustering will show the *unordered* client clusters converges as $\{\mathcal{S}_1^{(t)},\ldots,\mathcal{S}_K^{(t)}\} \to \{\mathcal{S}_1^*,\ldots,\mathcal{S}_K^*\}$ as $t\to\infty$. This directly implies that, $\hat{\boldsymbol{\theta}}_k^{(t)}$ will eventually align with the correct client cluster for the k-th distribution $\mathcal{D}_k$ as $t\to\infty$ and naturally the corresponding optimal model should be $\boldsymbol{\theta}_k^*$. Therefore, our model convergence analysis aims to show that $\hat{\boldsymbol{\theta}}_k^{(t)}$, the client plurality model for distribution $\mathcal{D}_k$ at iteration t, converges to $\boldsymbol{\theta}_k^*$, the optimal model for distribution $\mathcal{D}_k$ as $t\to\infty$.

Remark 1. Our algorithm differentiates itself by assuring convergence in both client clustering and model optimization, diverging from representative CFL approaches, which only guarantee to obtain the near-optimal models (Ghosh et al., 2020). These guarantees enable effective learning of clusters and models, circumventing the need for advantageous initial conditions such as partially well-clustered clients or near-optimal starting models, often assumed in existing research (Ghosh et al., 2020; Vahidian et al., 2023).

We begin by introducing some additional notation and assumptions. The normalized size of the smallest cluster is denoted by ρ as $\rho = \min_k |\mathcal{S}_k^*|/C$. To allow for theoretical analysis we shall assume that each iteration t each client c has access to an independent minibatch dataset $\mathcal{X}_c^{(t)}$ of size b from its associated data distribution.

Assumption 1. For each $k\in[K]$, we assume that there are at least two clients that have the distribution $\mathcal{D}_k$ – this implies $C\rho \geq 2$. We also assume that $Kd \geq C \geq (\log Kd)^{\frac{3}{2}}$

and $\|\boldsymbol{\theta}\| \leq \omega$ almost surely where $\omega > 0$, i.e., the norm of the model is bounded.

Discussions on Gaussianity of SGN. Several studies (Ahn et al., 2012; Chen et al., 2014; Jastrzębski et al., 2017; Hu et al., 2017; Jastrzębski et al., 2017; Zhu et al., 2019) have reached the conclusion that a Gaussian approximation for SGN is theoretically well-founded when the Central Limit Theorem applies and the size of minibatches is sufficiently large. Counterarguments have also been made suggesting that the Gaussian assumption may fail with the SGN being better modeled by a heavy-tailed distribution (Simsekli et al., 2019). Recently, Panigrahi et al. (Panigrahi et al., 2019) explored the statistical properties of SGN for deep neural networks for machine learning applications for numerous practical settings. They found that the SGN exhibits clear Gaussian characteristics during early phases of training and loses Gaussianity as learning progresses. This series of works suggests that at least in the early phases of learning, it may be realistic to assume it follows a Gaussian distribution for machine learning applications with neural networks. Thus, we make the following assumption on the SGN.

Assumption 2 (Gaussianity and bounded variance)**.** Let $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_b\}$ denote a dataset generated from $\mathcal{D}_k$ for some $k \in [K]$. For any given $\boldsymbol{\theta}$, we assume that $\mathbf{e}_c(\boldsymbol{\theta}) \sim \mathcal{N}(0, \boldsymbol{\Sigma})$, where $\mathcal{N}$ is multivariate Gaussian distribution with the covariance matrix $\boldsymbol{\Sigma}$ such that $\text{Tr}(\boldsymbol{\Sigma}) \leq v^2/b$ and $\|\boldsymbol{\Sigma}\|_\infty \leq \text{Tr}(\boldsymbol{\Sigma})$. For a single data instance $\mathbf{x} \sim \mathcal{D}_k$, we have $\mathbb{E}\|\nabla f(\{\mathbf{x}\}, \boldsymbol{\theta}) - \nabla F(\mathcal{D}_k, \boldsymbol{\theta})\|^2 \leq v^2$ and all SGN vectors, $\mathbf{e}_c(\boldsymbol{\theta})$, are independent.

Next, let $\mathcal{T}_k(t)$ denote the set of time indices where the k-th block of the feature vectors is updated before time t. Subsequently, the sequence of updates of the k-th model before time t can be denoted as $\tilde{\Theta}(k,t) = \{\boldsymbol{\theta}_k^{(t')} | t' \in \mathcal{T}_k(t)\}$.

Assumption 3 (Minimum average gradient gap)**.** For any k_1,$k_2 \in [K], k_1 \neq k_2$, a gap between average gradients over a sequence, which is given as $\min_{k,t} \| \sum_{\boldsymbol{\theta} \in \tilde{\Theta}(k,t)} \frac{1}{|\tilde{\Theta}(k,t)|} (\nabla F(\mathcal{D}_{k_1}, \boldsymbol{\theta}) - \nabla F(\mathcal{D}_{k_2}, \boldsymbol{\theta}))\|$ where $k \in [K]$ and $t > 1$, is greater than $\Delta_g (> 0)$, almost surely.

Assumption 3 is indeed a mild assumption for real-world problems as it states that the average gradients associated with different data distributions are different.

Now we present an analysis of the proposed algorithm from the following perspectives: (1) clustering performance, (2) contractive property, and (3) convergence.

(1) Clustering Performance

Let $W(t)$ denote the fraction of incorrectly clustered clients at t. The following Lemma 1 establishes the monotonic decrease of an upper bound of $W(t)$ as t increases, with its proof provided in Appendix B.1.

Lemma 1. *Consider any $\delta \in (0,1)$. Under Assumptions 1, 2, 3, and for t and T_{cl} such that $T_{cl} > t > KP+1$, the fraction of incorrectly clustered clients at t, $W(t)$, is upper bounded as $W(t) \leq \mathcal{O}(\frac{1}{\delta^2 t})$ with probability at least $1-\delta$.*

Remark 2 (Monotonic decrease of bound on number of incorrectly clustered clients)**.** The high probability result of the monotonic decrease in the upper bound of the number of incorrectly clustered clients is in stark contrast to the existing CFL algorithms, which do not guarantee enhanced clustering accuracy through an increase in the number of iterations (Ghosh et al., 2020; Sattler et al., 2020; Mansour et al., 2020). Instead, the clustering accuracy of the existing methods relies on large batch sizes or significant differences between clusters, which are often beyond the control of learning system operators.

(2) Contractive property

In Appendix B.2, we will show the following Theorem 1.

Assumption 4. The population loss function $F(\mathcal{D}_k, \boldsymbol{\theta})$ is μ-strongly convex and L-smooth for all $k \in [K]$. Recall that a differentiable function f is μ-strong convex if , $\forall \boldsymbol{\theta}', \boldsymbol{\theta}$, $f(\boldsymbol{\theta}') \geq f(\boldsymbol{\theta}) + \langle \nabla f(\boldsymbol{\theta}), \boldsymbol{\theta}' - \boldsymbol{\theta} \rangle + \frac{\mu}{2}\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|^2$ and L-smooth if $\forall \boldsymbol{\theta}', \boldsymbol{\theta}$, $\|\nabla f(\boldsymbol{\theta}') - \nabla f(\boldsymbol{\theta})\| \leq L\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|$.

Theorem 1. *Suppose that Assumptions 1-4 hold. Consider $\delta \in (0,1)$, $\gamma_k^{(t)} = \frac{2}{C(\mu+L)}$, and t such that $T_{cl} > t > \frac{\lambda K^2 PCv^2}{\delta^2 \Delta_g^2 \rho b} + KP + 1$ where λ is a constant.*

For any k, Algorithm 1 satisfies the following contractive property for the client plurality model of $\mathcal{D}_k$ with probability at least $1-\delta$.

$$\|\hat{\boldsymbol{\theta}}_k^{(t+1)} - \boldsymbol{\theta}_k^*\| \leq \sqrt{1 - \frac{3\rho\mu L}{(\mu+L)^2}} \|\hat{\boldsymbol{\theta}}_k^{(t)} - \boldsymbol{\theta}_k^*\| + \epsilon^{(t)} \quad (4)$$

$$\epsilon^{(t)} = \frac{6v}{\delta(\mu+L)\sqrt{bC\rho(1 - \frac{\lambda KCv^2}{4\rho\Delta_g^2\delta^2\lfloor \frac{t-1}{KP} \rfloor b})}} + \frac{\lambda LwKCv^2}{(\mu+L)\Delta_g^2\delta^2\lfloor \frac{t-1}{KP} \rfloor b} + \frac{3\sqrt{\lambda}K^2 v^2}{(\mu+L)\delta^2\Delta_g\sqrt{\lfloor \frac{t-1}{KP} \rfloor b}}. \quad (5)$$

Theorem 1 shows a contractive property of CFL-GP with a per-iteration error rate $\epsilon^{(t)}$ as in (5). We note that this error rate is asymptotically optimal.

Remark 3 (Optimality of error rate)**.** When the CU in the CFL framework knows all clients' distribution identities, the *optimal error rate* $\epsilon^{(t)}$ in (5) with respect to the batch size b and number of clients C is shown to be $\tilde{\mathcal{O}}(1/\sqrt{bC})$ (Ghosh et al., 2020), where the order notation $\tilde{\mathcal{O}}$ omits the other parameters except b and C. It is worth noting that the order

The constant is not problem-specific and given in B.2.1.

of error rate $\epsilon^{(t)}$ of CFL-GP converges to this one as $t \to \infty$. We also note that the guaranteed increase in clustering accuracy shown in Lemma 1 results in a monotonic decrease of error rate with respect to t, demonstrating that the model update towards the optimal solution becomes increasingly accurate with more iterations.

Notably, the contractive property is guaranteed without requiring any good initialization assumptions, unlike prior works that require the initial model to be near the optimal, a correct initial clustering of certain amounts of clients (Ghosh et al., 2020; Vahidian et al., 2023), or rely on clustering triggering parameters that potentially require prior knowledge of the gradient statistics (Sattler et al., 2020).

(3) Convergence

In Appendix B.4, we will show the following Proposition 1.

Proposition 1. *Suppose Assumptions 1-4 hold and for all* c, k, t, $\|\nabla f(\mathcal{X}_c^{(t)}, \boldsymbol{\theta}_k^{(t)})\| \leq H$ *almost surely. Consider any* $\delta \in (0,1)$, $\gamma_k^{(t)} = 1/(\mu|\mathcal{S}_k^{(t)}|(\max(t - T_{cl} + 1, 1)))$, *and* T *such that* $T \geq T_{cl} + 4$ *with* $T_{cl} > (\frac{\lambda K C^2 v^2}{\delta^2 \Delta_g^2 b} + 1)KP + 1$.

For any k, *after* T *iterations, Algorithm 1 satisfies the following convergence property for the client plurality model of* $\mathcal{D}_k$ *with probability at least* $1 - \delta$.

$$\|\hat{\boldsymbol{\theta}}_k^{(T)} - \boldsymbol{\theta}_k^*\| \leq \mathcal{O}\big(\frac{\log(\log(T - T_{cl})/\delta)}{T}\big). \tag{6}$$

Proposition 1 indicates that the proposed algorithm can achieve the order of the conventional stochastic gradient (Rakhlin et al., 2012) and the client plurality model for the k-th distribution, $\hat{\boldsymbol{\theta}}_k$, converges to the optimal model $\boldsymbol{\theta}_k^*$ as $T \to \infty$ with high probability.

5. Numerical Evaluation

Evaluation Summary and Roadmap. We shall show the superiority of CFL-GP over a comprehensive set of state-of-the-art baselines on various benchmarks. Below we provide an overview, highlighting the key themes of each experiment.

[Section 5.1]: We conduct a thorough assessment of our algorithm on a linear regression task and its *robustness* to environmental variables such as client count, batch size, data distribution heterogeneity, and model initialization, utilizing controllable synthetic datasets. **[Section 5.2]**: We then extend our evaluation to neural network based image classification tasks, again examining *robustness* to client number and batch size, data distribution heterogeneity, as well as *communication and computation costs*. **[Section 5.3]**: We further evaluated our proposed algorithm on large-scale models (e.g., ResNet-18, deep autoencoders with over a million parameters) with image and non-image industrial datasets and assessed its *applicability for high-dimensional models and diverse datasets*. **[Appendix D]**: This appendix includes a detailed analysis of *computation and communication costs*, including clustering convergence speed and accuracy, is provided. **[Appendix F]**: details the configuration of all simulation environments and algorithms, and includes additional ablation studies for the proposed algorithm. **[Appendicies F.2 and F.3]**: In these sections, we compare CFL-GP's performance with recent algorithms in related domains, such as PFL and other CFL algorithms from different communication protocols.

Across varied setups, CFL-GP consistently outperforms in task and clustering performance, demonstrating its robustness and effectiveness.

Baselines. We evaluate representative theoretically well-founded CFL algorithms, including: Iterative Federated Clustering Algorithm (IFCA) (Ghosh et al., 2020), Model-Agnostic Distributed Multi-Task Optimization (MADMO) (Sattler et al., 2020), and Principal Angles analysis for Clustered Federated Learning (PACFL) (Vahidian et al., 2023), as well as traditional FL, which uses a single global model (McMahan et al., 2017). In Sec. 5, we primarily focus on contrasting the algorithms CFL-GP, IFCA, MADMO, and the traditional approach using a single model due to their comparable FL communication protocols: gradient sharing (CFL-GP, IFCA, MADMO, Global Model) and model evaluation sharing (IFCA). PACFL, using sharing of principal angle analysis of dataset to the CU, is analyzed separately in Appendix F.3. We also discuss PFL methods such as FedEM (Marfoq et al., 2021), noting their differences from CFL, with further comparisons in Appendix F.2.

Metric. The performance of CFL can be evaluated using clustering accuracy and overall loss. To measure clustering accuracy, we use the Adjusted Rand Index (ARI) (Hubert & Arabie, 1985; Steinley, 2004), which ranges from -0.5 to 1.0. An ARI of 1.0 indicates optimal clustering, while an ARI of 0.0 indicates random clustering. We also provide a comparison of computation and communication costs.

5.1. Synthetic Dataset: Multiple Linear Regressions over Gaussian Label Noise

We first consider a CFL problem where the objective is to learn three linear models for regression tasks. For a given $k \in [3]$, we generate data pairs $\mathbf{x} = (x, y)$ as $y = x\tan(\phi_k) + n$ where $n \sim \mathcal{N}(0, 0.2^2)$ and x follows $\mathcal{U}(0, \cos(\phi_k))$, a uniform distribution over $[0, \cos(\phi_k)]$. We consider three angles (ϕ_1, ϕ_2, ϕ_3) = ($\Delta\phi$, 0, $-\Delta\phi$) for vaious $\Delta\phi$s. The k-th model's output is computed as $\hat{y} = \boldsymbol{\theta}_{k,1}x + \boldsymbol{\theta}_{k,2}$ with two initialization strategies for $\boldsymbol{\theta}_{k,1}^{(0)}$, $\boldsymbol{\theta}_{k,1}^{(0)} \sim \mathcal{U}(-0.8, 0.8)$ and $\boldsymbol{\theta}_{k,1}^{(0)} \sim \mathcal{U}(-1.6, 1.6)$. In all cases, $\boldsymbol{\theta}_{k,2}^{(0)} = 0$. See Appendix F.4 for further details.

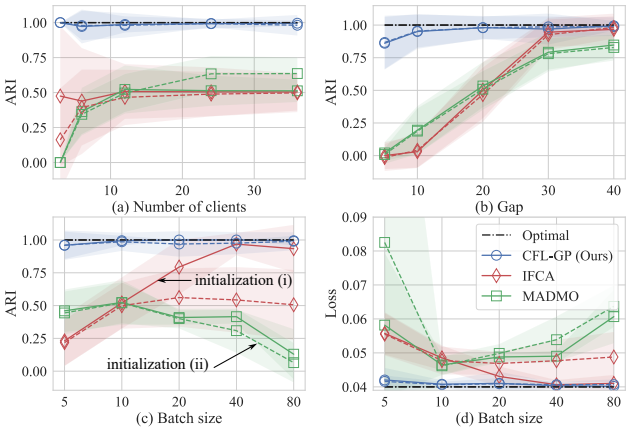

Figure 1. Multiple Linear Regression: The clustering performance (ARI) according to the number of clients C, data distribution gap $\Delta\phi$, batch size b, and initialization strategies, respectively. Subplot (d) shows the corresponding Loss values (MSE) for Subplot (c). CFL-GP exhibits *near-optimal* performance and is much less sensitive to the environmental variables and model gap/initialization than the existing CFL algorithms.

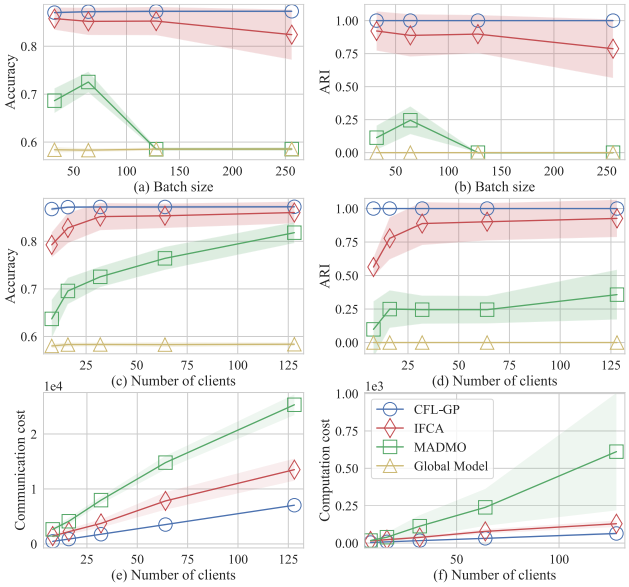

Figure 2. Rotated MNIST: Accuracy and ARI are plotted against batch size and number of clients in subplots (a)-(d), while communication and computation costs are plotted in subplots (e) and (f), respectively. CFL-GP's rapid and robust clustering (b,d) not only lowers communication and computational costs to reach target performance (e,f) but also yields high final accuracy (a,c).

Robustness of CFL-GP. In Figure 1 we plot various CFL algorithms' ARI in relation to the number of clients, cluster gaps, and batch sizes in subplots (a), (b), and (c) and the corresponding task loss (MSE) in (d). The solid lines correspond to the first initialization strategy mentioned above, while the dotted lines correspond to the other (default setup with $b=10$, $\Delta\phi=20°$, and $C=12$). Throughout the results in (a) to (c), we observe that the CFL-GP exhibits impressive performance with an *ARI close to 1.0*, being robust over the system parameters and outperforming the baselines, leading to *near-optimal task performance* in (d). The results in (b) especially demonstrate the statistical rigor and reliability of CFL-GP's clustering; when $\Delta\phi = 5°$, the clustering performance of both baselines is equivalent to random clustering (0 ARI), while CFL-GP achieves an ARI above 0.8. The results in (c) show that CFL-GP is much more robust over the *initialization* of models than others.

5.2. Real Image Dataset: Rotated MNIST

This subsection presents an extension to the benchmark commonly used in CFL literature (Lopez-Paz & Ranzato, 2017; Ghosh et al., 2020; Sattler et al., 2020). We employ the MNIST dataset (LeCun et al., 1998), which contains handwritten digits with 10 classes. The dataset is divided equally among eight sets of clients, and rotation transformations of 0, 15, 90, 105, 180, 195, 270, and 275 degrees are applied to each client's data ($D = 8$). Consequently, the clients possess heterogeneous datasets due to the division and rotation transformations.

Our goal is to train four nonlinear models ($K = 4$), each with over 150K parameters, to accurately classify the digits in the transformed datasets without knowledge of the cluster identities (specific rotation applied to the data). We assess the results using the ARI, assuming that optimal partitioning occurs when clusters with similar rotation angles are grouped together. For a detailed experimental setup, see Appendix F.5.

Classification accuracy and clustering performance. Subplots (a-d) in Figure 2 present the average classification accuracy and ARI after $T = 200$ of the algorithms with respect to changes in batch size and number of clients. CFL-GP demonstrates remarkable consistency in achieving perfect clustering performance with ARI scores of 1.0 across various batch sizes and client numbers, as illustrated in Figure 2-(b,d). This allows for the efficient allocation of similar data distributions to each model, resulting in the highest classification accuracy observed in all experiments (a,c). Competing CFL algorithms, on the other hand, exhibit sensitivity to both batch sizes and client numbers, with their ARI and accuracy patterns fluctuating significantly as a result (a-d). Notably, these algorithms experience substantial ARI degradation, especially when applied to small client numbers (d).

Communication and computation cost. Subplot (e) in

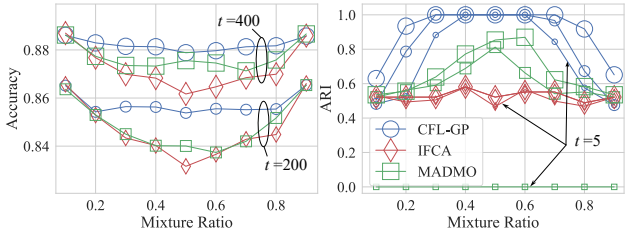

Figure 3. Accuracy and ARI vs. mixture ratio κ. CFL-GP achieves higher model performance (classification accuracy, left) as well as higher clustering accuracy (ARI, right) across all mixture ratios. CFL-GP's ARI at $t = 5$ exceeds that of others at $t = 400$ in most cases, demonstrating CFL-GP's fast clustering ability (right).

Table 1. Performance comparison in various benchmarks including Industrial and Large-Scale scenarios, demonstrating CFL-GP's versatility with consistent attainment of optimal ARI of 1.

Algorithm	[Exp 1: Deep AE / COST2100] NMSE/ARI $C=16, C=32$		
CFL-GP	**-19.38(dB)±3e-3/1.00±0.00**	**-17.56(dB)±7e-3/1.00±0.00**	
IFCA	-15.83(dB)±6e-3/-0.01±0.01	-14.07(dB)±1e-2/0.00±0.00	
MADMO	-17.63(dB)±1e-2/0.50±0.50	-16.28(dB)±2e-3/0.44±0.18	
FedAvg	-14.57(dB)±6e-3/0.00±0.00	-14.47(dB)±3e-3/0.00±0.00	
Algorithm	[Exp 2:Resnet18, CIFAR10] ACC/ARI for $C=20$, $C=40$, $C=80$		
CFL-GP	**73.93±0.50/1.00±0.00**	**72.16±0.33/1.00±0.00**	**70.49±0.35/1.00±0.00**
IFCA	**73.97±0.37/1.00±0.00**	**72.15±0.33/1.00±0.00**	**70.32±0.32/1.00±0.00**
MADMO	68.42±2.14/0.27±0.27	65.56±2.36/0.14±0.17	52.53±7.44/0.00±0.00
FedAvg	70.75±1.10/0.00±0.00	65.59±2.91/0.00±0.00	52.48±8.15/0.00±0.00
Algorithm	[Exp 3: CNN, EMNIST] ACC/ARI for $C=10$, $C=80$, $C=160$		
CFL-GP	**82.28±0.30/1.00±0.00**	**79.11±0.34/1.00±0.00**	**76.73±0.24/1.00±0.00**
IFCA	78.68±5.41/0.71±0.45	77.52±4.62/0.86±0.35	**76.55±0.21/1.00±0.00**
MADMO	76.73±5.89/0.53±0.47	78.30±0.52/0.74±0.09	75.07±0.52/0.40±0.15
FedAvg	70.09±0.78/0.00±0.00	65.77±0.56/0.00±0.00	66.23±0.71/0.00±0.00

Figure 2 displays the required communication cost (total number of model transmissions between CU and all clients) and computation cost (total runtime for clustering, gradient computation, and forward computing) to achieve an accuracy of 0.6 (achievable by all the CFL algorithms), as a function of the number of clients. In all cases, CFL-GP achieves the lowest computation and communication costs owing to its rapid convergence towards optimal clustering, which facilitates fast and precise model learning. For a more detailed analysis of the learning process, see Appendix F.5. We also note that CFL-GP has the lowest runtime and communication cost for a given round on average due to its rapid convergence (Appendix D).

Extension to mixture distributions and convergence. To evaluate the robustness of the CFL-GP based on distribution similarity, the dataset is partitioned into three groups: one rotated 0 degrees, another 180 degrees, and the remaining group is subjected to the two rotation transformations with probabilities κ and $1-\kappa$, respectively. The data is divided equally among four clients for each group. See Appendix F.5 for more details.

Figure 3 (Right) illustrates the changes in ARI for $t = 5, 200, 400$ and corresponding accuracy as a function of κ, with larger markers indicating greater t. CFL-GP's ARI at $t = 5$ outperforms that of all other algorithms at $t=400$ when $\kappa \in [0.3, 0.7]$, and it achieves the highest ARI score at $t = 400$. Also, CFL-GP achieves optimal clustering at $t=5$, particularly when κ was around 0.5, which is a relatively easy mixture of distributions to cluster.

Figure 3 (Left) shows that CFL-GP's rapid convergence to optimal clustering leads to improved accuracy for a given mixture ratio κ and t. When κ is close to zero or one, the average accuracy slightly improves as clients with a more biased mixture of distributions have better performance without confusion from rotation transformations.

5.3. Additional Experiments Including Large Scale and Industrial Applications

We demonstrate the robustness of CFL-GP on various practical scenarios and large scale applications, an overview of which is provided in the following, along with part of the results presented in Table 1.

In **[Exp 1]**, we address a Channel State Information (CSI) compression problem via deep AutoEncoders (AE) with wireless channel datasets such as COST2100 (Wen et al., 2018) and (Jaeckel et al., 2021). CFL algorithms are required to cluster local data centers having heterogeneous data distributions where they typically exhibit high sample variance, posing challenges in distinguishing them. In **[Exp 2]**, the aim is to cluster clients with a part of the CIFAR10 dataset (Krizhevsky et al., 2009) and a large neural network ResNet18 (He et al., 2016) based on their label heterogeneity while allowing partial model sharing among the client clusters. In **[Exp 3]**, we adopt a setup from (Sattler, 2020) that involves client clustering where clients have a part of the EMNIST dataset (Cohen et al., 2017) with both heterogeneous label distribution and rotation transformation.

Notably, the CFL-GP algorithm consistently achieves the perfect clustering (ARI score of 1.0) and the highest performance across all tasks, outperforming other approaches by a significant margin, both in terms of the clustering accuracy and the model performance. Although not shown in the table, we note that CFL-GP has the lowest average runtime in most simulations resulting from its rapid convergence. (See Appendix F for details).

6. Conclusion and Discussion

We have theoretically established and validated that CFL clients' model gradients during the learning process can be

used to effectively group clients, delivering excellent results. Indeed our proposed CFL-GP algorithm is able to circumvent the challenges associated with high dimensionality and noise while preserving clustering and learning optimality. Our algorithm not only satisfies theoretical guarantees under mild assumptions but also demonstrates significantly faster clustering performance, up to 100 times faster than the state-of-the-art, leading to better performance at a lower cost. The computational and communication costs can further be reduced by exploiting highly compressed gradients with CFL-GP still preserving optimality for a wide range of tasks.

In practical scenarios with uncontrollable variables and unknown data statistics, a CFL algorithm's performance robustness to hyperparameter choices is crucial. Unlike MADMO and PACFL, CFL-GP does not require cluster-branching sensitivity hyperparameters, avoiding potentially inconsistent outputs. Instead, CFL-GP and IFCA require the fixing of the number of models to be learned, and we propose a method in Appendix G to efficiently identify what it should be in practical settings. This highlights CFL-GP's adaptability to various learning contexts.

Acknowledgements

We thank the anonymous reviewers for their constructive comments. This work was partly supported by InterDigital, INC. through 6G@UT center within the Wireless Networking and Communications Group (WNCG) at the University of Texas at Austin, ARO Award W911NF2310062, ONR Award N00014-21-1-2379, National Science Foundation under grant no. 2148224, NSF Award CNS-2008824, funds from OUSD R&E, NIST, and industry partners as specified in the Resilient & Intelligent NextG Systems (RINGS) program.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Aghajanyan, A., Zettlemoyer, L., and Gupta, S. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

Ahn, S., Balan, A. K., and Welling, M. Bayesian posterior sampling via stochastic gradient fisher scoring. In *International conference on machine learning*. PMLR, 2012.

Arthur, D. and Vassilvitskii, S. K-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, 2007.

Bandeira, A. S., Boedihardjo, M. T., and van Handel, R. Matrix concentration inequalities and free probability. *arXiv preprint arXiv:2108.06312*, 2021.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečnỳ, J., Mazzocchi, S., McMahan, B., et al. Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems*, 1:374–388, 2019.

Caruana, R. Multitask learning. *Machine learning*, 28(1): 41–75, 1997.

Chen, T., Fox, E., and Guestrin, C. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pp. 1683–1691. PMLR, 2014.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Collobert, R. and Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 160–167, 2008.

Corinzia, L., Beuret, A., and Buhmann, J. M. Variational federated multi-task learning. *arXiv preprint arXiv:1906.06268*, 2019.

Duan, M., Liu, D., Ji, X., Wu, Y., Liang, L., Chen, X., Tan, Y., and Ren, A. Flexible clustered federated learning for client-level data distribution shift. *IEEE Transactions on Parallel and Distributed Systems*, 2021.

Fallah, A., Mokhtari, A., and Ozdaglar, A. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Ghosh, A., Hong, J., Yin, D., and Ramchandran, K. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*, 2019.

Ghosh, A., Chung, J., Yin, D., and Ramchandran, K. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33: 19586–19597, 2020.

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

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021a.

Hu, W., Li, C. J., Li, L., and Liu, J.-G. On the diffusion approximation of nonconvex stochastic gradient descent. *arXiv preprint arXiv:1705.07562*, 2017.

Hu, Y., Yang, W., Ma, Z., and Liu, J. Learning end-to-end lossy image compression: A benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021b.

Hubert, L. and Arabie, P. Comparing partitions. *Journal of classification*, 2:193–218, 1985.

Jaeckel et al., S. Quadriga - quasi deterministic radio channel generator, user manual and documentation. *Fraunhofer Heinrich Hertz Institute*, 2.6.1, 2021.

Jastrzębski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

Jiang, J. Image compression with neural networks–a survey. *Signal processing: image Communication*, 14(9):737–760, 1999.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Kim, H. Github repository for clustered federated learning via gradient-based partitioning. https://github.com/Heasung-Kim/clustered-federated-learning-via-gradient-based-partitioning, 2024.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Konečnỳ, J., McMahan, H. B., Ramage, D., and Richtárik, P. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

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

Kulkarni, V., Kulkarni, M., and Pant, A. Survey of personalization techniques for federated learning. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pp. 794–797. IEEE, 2020.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Li, C., Farkhoor, H., Liu, R., and Yosinski, J. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

Lin, X. An overview of 5g advanced evolution in 3gpp release 18. *IEEE Communications Standards Magazine*, 6(3):77–83, 2022. doi: 10.1109/MCOMSTD.0001.2200001.

Löffler, M., Zhang, A. Y., and Zhou, H. H. Optimality of spectral clustering in the gaussian mixture model. *The Annals of Statistics*, 49(5):2506–2530, 2021.

Long, G., Xie, M., Shen, T., Zhou, T., Wang, X., and Jiang, J. Multi-center federated learning: clients clustering for better personalization. *World Wide Web*, 26(1):481–500, 2023.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2016.

Lu, Z., Wang, J., and Song, J. Multi-resolution csi feedback with deep learning in massive mimo system. In *IEEE International Conference on Communications*, pp. 1–6. IEEE, 2020.

Ma, S., Zhang, X., Jia, C., Zhao, Z., Wang, S., and Wang, S. Image and video compression with neural networks: A review. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(6):1683–1698, 2019.

Mansour, Y., Mohri, M., Ro, J., and Suresh, A. T. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

Marfoq, O., Neglia, G., Bellet, A., Kameni, L., and Vidal, R. Federated multi-task learning under a mixture of distributions. *Advances in Neural Information Processing Systems*, 34:15434–15447, 2021.

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

Montavon, G., Samek, W., and Müller, K.-R. Methods for interpreting and understanding deep neural networks. *Digital signal processing*, 73:1–15, 2018.

Niknam, S., Dhillon, H. S., and Reed, J. H. Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Communications Magazine*, 58(6):46–51, 2020.

Panigrahi, A., Somani, R., Goyal, N., and Netrapalli, P. Non-gaussianity of stochastic gradient noise. In *Science meets Engineering of Deep Learning (SEDL) Workshop, 33rd Conference on Neural Information Processing Systems*, 2019.

Rakhlin, A., Shamir, O., and Sridharan, K. Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1571–1578, 2012.

Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H. R., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K., et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020.

Ruan, Y. and Joe-Wong, C. Fedsoft: Soft clustered federated learning with proximal local updating. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8124–8131, 2022.

Ruder, S. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

Sattler, F. Clustered federated learning: Model-agnostic distributed multi-task optimization under privacy constraints. https://github.com/felisat/clustered-federated-learning, 2020.

Sattler, F., Müller, K.-R., and Samek, W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.

Shamir, O. A stochastic pca and svd algorithm with an exponential convergence rate. In *International conference on machine learning*, pp. 144–152. PMLR, 2015.

Shamir, O. Fast stochastic algorithms for svd and pca: Convergence properties and convexity. In *International Conference on Machine Learning*, pp. 248–256. PMLR, 2016.

Shenaj, D., Fanì, E., Toldo, M., Caldarola, D., Tavera, A., Michieli, U., Ciccone, M., Zanuttigh, P., and Caputo, B. Learning across domains and devices: Style-driven source-free domain adaptation in clustered federated learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 444–454, 2023.

Simsekli, U., Sagun, L., and Gurbuzbalaban, M. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning*, pp. 5827–5837. PMLR, 2019.

Singhal, K., Sidahmed, H., Garrett, Z., Wu, S., Rush, J., and Prakash, S. Federated reconstruction: Partially local federated learning. *Advances in Neural Information Processing Systems*, 34:11220–11232, 2021.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. Federated multi-task learning. *Advances in neural information processing systems*, 30, 2017.

Steinley, D. Properties of the hubert-arable adjusted rand index. *Psychological methods*, 9(3):386, 2004.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Torrey, L. and Shavlik, J. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pp. 242–264. IGI global, 2010.

Vahidian, S., Morafah, M., Wang, W., Kungurtsev, V., Chen, C., Shah, M., and Lin, B. Efficient distribution similarity identification in clustered federated learning via principal angles between client data subspaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10043–10052, 2023.

Vardhan, H., Ghosh, A., and Mazumdar, A. An improved federated clustering algorithm with model-based clustering. *Transactions on Machine Learning Research*, 2024.

Vogels, T., Karimireddy, S. P., and Jaggi, M. Powersgd: Practical low-rank gradient compression for distributed

optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Wang, K., Mathews, R., Kiddon, C., Eichner, H., Beaufays, F., and Ramage, D. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019.

Wang, T., Wen, C.-K., Jin, S., and Li, G. Y. Deep learning-based csi feedback approach for time-varying massive mimo channels. *IEEE Wireless Communications Letters*, 8(2):416–419, 2018.

Wen, C.-K., Shih, W.-T., and Jin, S. Deep learning for massive mimo csi feedback. *IEEE Wireless Communications Letters*, 7(5):748–751, 2018.

Xu, J., Tong, X., and Huang, S.-L. Personalized federated learning with feature alignment and classifier collaboration. *International Conference on Learning Representations*, 2023.

Yan, Y., Tong, X., and Wang, S. Clustered federated learning in heterogeneous environment. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Yang, L., Tan, B., Zheng, V. W., Chen, K., and Yang, Q. Federated recommendation systems. In *Federated Learning*, pp. 225–239. Springer, 2020.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

Yu, T., Bagdasaryan, E., and Shmatikov, V. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020.

Zeng, D., Hu, X., Liu, S., Yu, Y., Wang, Q., and Xu, Z. Stochastic clustered federated learning. *arXiv preprint arXiv:2303.00897*, 2023.

Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., and Gao, Y. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

Zhu, Z., Wu, J., Yu, B., Wu, L., and Ma, J. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. In *International Conference on Machine Learning*, pp. 7654–7663. PMLR, 2019.

Contents

1 Introduction **1**

2 System Model **2**

3 Algorithm **2**

4 Theoretical Analysis **4**

5 Numerical Evaluation **6**
5.1 Synthetic Dataset: Multiple Linear Regressions over Gaussian Label Noise . . . 6
5.2 Real Image Dataset: Rotated MNIST . . . 7
5.3 Additional Experiments Including Large Scale and Industrial Applications . . . 8

6 Conclusion and Discussion **8**

A Related Work **14**

B Technical Results **15**
B.1 Proof of Lemma 1 . . . 16
B.2 Proof of Theorem 1 . . . 20
B.3 Proof of Lemma 3 . . . 27
B.4 Proof of Proposition 1 . . . 30

C Extensions for Improving Communication Efficiency **30**

D Computation and Communication Cost **31**
D.1 Convergence speed towards optimal clustering . . . 31
D.2 Communication and computation cost before clustering convergence . . . 33

E Compressed or selected gradient information **34**
E.1 Compressed gradient information . . . 34
E.2 Selected gradient information . . . 37

F Experiment Results **38**
F.1 Ablation studies for feature moving average factor β . . . 39
F.2 Comparison with Personalized Federated Learning utilizing implicit clustering . . . 39
F.3 Comparison with recent CFL algorithms from different communication protocols . . . 40
F.4 Experiments on Multiple Linear Regression Tasks . . . 41
F.5 Experiments on MNIST Dataset . . . 42
F.6 Experiments on Wireless Channel Datasets . . . 47
F.7 Experiments on CIFAR10 and Resnet18 . . . 50
F.8 Experiments on EMNIST Dataset . . . 52
F.9 Summary of experiment results . . . 53

G Dealing with Unknown Number of Clusters **53**

A. Related Work

Federated Learning. FL is a distributed learning framework in which multiple clients with local datasets participate in training to learn a global model without sharing the raw local dataset (Konečnỳ et al., 2016; McMahan et al., 2017; Li et al., 2019; 2020; Zhang et al., 2021); success has been shown across various fields throughout the industry (Bonawitz et al., 2019; Yang et al., 2020; Niknam et al., 2020; Rieke et al., 2020). However, one of the main challenges of FL is the presence of non-i.i.d. data across clients (Kairouz et al., 2021), which implies that applying one global model to all clients under the traditional FL framework may not be the optimal solution for the clients.

Personalized Federated Learning. In order to overcome the data heterogeneity across the clients due to the non-i.i.d. nature, research on FL system design for model personalization draws valuable attention (Kulkarni et al., 2020). Traditional model personalization involves multiple clients cooperating to train a global model and transforming that global model to fit each client's unique local dataset. In (Wang et al., 2019), after training one global model, the authors presented a method of additional learning starting with the global model using the local data of each client, which can be interpreted as a transfer learning technique (Torrey & Shavlik, 2010). A model-agnostic meta learning (Finn et al., 2017) has been applied to the FL framework (Fallah et al., 2020) to find an initial global model that allows clients to quickly fine-tune the model with a local dataset. Partially Local Federated Learning (Singhal et al., 2021) is also motivated by the meta learning, which shows fast personalization of the global model. Multitask learning (MTL) frameworks (Caruana, 1997) for learning personalized models are studied under the FL framework in (Smith et al., 2017; Corinzia et al., 2019; Marfoq et al., 2021).

Recently, (Ruan & Joe-Wong, 2022) especially considered mixture distribution scenario with focus on learning a global model for future adaptation and multiple personalized models. To enhance feature representation, recent work (Xu et al., 2023) introduced an approach that leverages feature alignment and classifier combination to address personalization challenges within deep neural networks. Ruan et al. (2022) addressed scenarios characterized by mixture distributions, focusing on the development of a global model primed for future adaptation alongside several personalized models. (Xu et al., 2023) introduced a method that capitalizes on feature alignment and classifier combinations to address the challenges of personalization in deep neural networks.

Clustered Federated Learning. CFL can be distinguished from Personalized Federated Learning (PFL) in that CFL considers the *absence of a global model* for clients' datasets and desired tasks. CFL not only aims to provide a common model for each set of clients but also determines the appropriate cluster identity assignments for each participating client during the learning process. In many relevant practical tasks, it has been found that providing a shared model to clusters of clients is advantageous over providing a personalized model for each client; we discuss some motivating practical examples in Appendix F.6.

Under the CFL framework, (Sattler et al., 2020) introduced the bipartitioning-based CFL algorithm. The loss-based client clustering method was proposed in (Mansour et al., 2020; Ghosh et al., 2020) along with thorough theoretical analyses. In addition, the clustering method based on the locally trained empirical risk minimizer was proposed in (Ghosh et al., 2019). (Duan et al., 2021) considered distribution shift issues on the CFL framework with clustering based on optimization direction similarities. Recently, (Shenaj et al., 2023) proposed a clustering algorithm that utilizes clients' inference similarity, designed for scenarios where distinct user groups may have unique objectives yet can benefit from mutual leveraging through clustering. In (Vahidian et al., 2023), introduced a clustering method that operates by examining the principal angles between client data subspaces.

(Long et al., 2023) devised an effective loss function, which incorporates a distance-based regularization term. This term penalizes deviations from central models, promoting convergence of local models towards a common objective. (Zeng et al., 2023) utilized cosine similarity of updates from an anchor model to construct a client similarity matrix, enabling effective client clustering. (Yan et al., 2023) employed cosine similarity of model parameters combined with the average connectivity within client clusters to optimize the clustering process. (Vardhan et al., 2024) proposed a new successive clustering algorithm for CFL that adapts clustering based on local model similarities, which does not require prior knowledge of the cluster count.

Note that our approach stands out from the existing methods by strengthening clustering criteria through the accumulation of gradient information acquired during the training process, which leads to theoretical optimality for both client clustering and model convergence, and empirical robustness.

Table 2. Notation and Description

Notation	Description	Note
$\mathbf{G}^{(t)}$	gradient profile matrix $\mathbf{G}^{(t)} = (\boldsymbol{g}_1^{(t)} \cdots \boldsymbol{g}_C^{(t)})$ at t	$\mathbf{G}^{(t)} \in \mathbb{R}^{Kd \times C}$
$\mathbf{E}^{(t)}$	stochastic gradient noise matrix at t	
$\mathbf{e}_c(\boldsymbol{\theta})$	stochastic gradient noise on model $\boldsymbol{\theta}$ of client c	
$\mathcal{X}_c^{(t)}$	minibatch (set of data instances) from client c at t	
$\mathbf{x}$	data instance	
K	total number of models (clusters)	
C	total number of clients	
P	clustering period	
$\mathcal{S}_k^{(t)}$	set of clients assigned to the k-th model at t	
$\mathcal{S}_k^*$	set of clients whose distribution identity is k	$\lvert\mathcal{S}_k^*\rvert > 0$
$\mathcal{S}_{\bar{k}}^*$	set of clients whose distribution identity is not k	
$\hat{\mathcal{S}}_k^{(t)}$	set of clients which the plurality of clients in $\mathcal{S}_k^*$ agree at t	
b	batch size	
d	number of model parameters	$\boldsymbol{\theta} \in \mathbb{R}^d$
k_c^*	Distribution identity for c-th client	$k_c^* \in [K]$
$k_c^{(t)}$	estimated cluster identity for c-th client at t	$k_c^{(t)} \in [K]$
$\boldsymbol{\theta}_k$	k-th model (a set of parameters)	$\boldsymbol{\theta}_k \in \mathbb{R}^d$
$\hat{\boldsymbol{\theta}}_k$	client plurality model associated with k-th distribution	
ρ	normalized minimum number of clients in cluster	$\rho = \min_c \frac{\lvert\mathcal{S}_k^*\rvert}{C}$
ω	maximum norm of $\boldsymbol{\theta}$	
Ψ	bijective permutation function space	$\Psi = \{\psi : [K] \xrightarrow{\mathsf{B}} [K]\}$
Π	surjective clustering function space	$\Pi = \{\pi : [C] \xrightarrow{\mathsf{S}} [K]\}$
Θ	model parameter space	$\Theta \subset \mathbb{R}^d$

Algorithm 2 CFL-GP

Input: K initial models $\{\boldsymbol{\theta}_k^{(0)}\}_{k=1}^K$, K initial clusters $\{\mathcal{S}_k^{(0)}\}_{k=1}^K$, clustering period P, learning rate $\gamma_k^{(t)}$, gradient features initialized as $\boldsymbol{g}_c^{(-1)} = \mathbf{0}\ \forall c \in [C]$, feature moving average factor $\{\beta_t\}_{t=1}^T$, and the number of cluster updates T_{cl}, broadcast model index $\bar{k}^{(-1)} = K$

Output: K trained models $\{\boldsymbol{\theta}_k^{(T)}\}_{k=1}^K$ and K updated clusters $\{\mathcal{S}_k^{(T)}\}_{k=1}^K$

for $t = 0$ **to** T **do**
  for $k = 1$ **to** K in parallel **do** ▷ *Model Update*
    CU transmits $\boldsymbol{\theta}_k^{(t)}$ to the clients in $\mathcal{S}_k^{(t)}$ and receives $\nabla f(\mathcal{X}_c^{(t)}, \boldsymbol{\theta}_k^{(t)})\ \forall c \in \mathcal{S}_k^{(t)}$
    CU updates $\boldsymbol{\theta}_k^{(t+1)} := \boldsymbol{\theta}_k^{(t)} - \gamma_k^{(t)} \sum_{c \in \mathcal{S}_k^{(t)}} \nabla f(\mathcal{X}_c^{(t)}, \boldsymbol{\theta}_k^{(t)})$
  if $t \bmod P = 1$ and $t < T_{\text{cl}}$ **then** ▷ *Cluster Update*
    Update $\bar{k}^{(t)} := (\bar{k}^{(t-1)} \bmod K) + 1$ ▷ round-robin manner
    CU broadcasts $\boldsymbol{\theta}_{\bar{k}}^{(t)}$ and receives $\nabla f(\mathcal{X}_c^{(t)}, \boldsymbol{\theta}_{\bar{k}}^{(t)})\ \forall c \in [C]$
    CU updates $\boldsymbol{g}_{c,\bar{k}}^{(t)} := (1 - \beta_t)\boldsymbol{g}_{c,\bar{k}}^{(t-1)} + \beta_t \nabla f(\mathcal{X}_c^{(t)}, \boldsymbol{\theta}_{\bar{k}}^{(t)})\ \forall c \in [C]$
    $\boldsymbol{g}_{c,k'}^{(t)} := \boldsymbol{g}_{c,k'}^{(t-1)}\ \forall c \in [C], k' \in [K]$ s.t. $k' \neq \bar{k}$
    $\{\mathcal{S}_k^{(t+1)}\}_{k=1}^K := \textsc{SpectralClustering}(\{\boldsymbol{g}_c^{(t)}\}_{c=1}^C, \{\mathcal{S}_k^{(t)}\}_{k=1}^K)$
  else
    $\mathcal{S}_k^{(t+1)} := \mathcal{S}_k^{(t)}\ \forall k \in [K]$, $\boldsymbol{g}_c^{(t)} := \boldsymbol{g}_c^{(t-1)}\ \forall c \in [C]$, and $\bar{k}^{(t)} := \bar{k}^{(t-1)}$

B. Technical Results

In preparation for the theoretical proofs to follow, Table 2 lists key notations, Algorithm 2 formalizes CFL-GP, including the notation for the broadcast model index at t as $\bar{k}^{(t)}$, and Algorithm 3 details the SPECTRALCLUSTERING subroutine.

Algorithm 3 SPECTRALCLUSTERING

Input: gradient features $\{\boldsymbol{g}_c^{(t)}\}_{c=1}^C$, client clusters $\{\mathcal{S}_k^{(t)}\}_{k=1}^K$ ▷ $\mathbf{G}^{(t)} = (\boldsymbol{g}_1^{(t)}, \cdots, \boldsymbol{g}_C^{(t)})$

Output: ordered client clusters $(\{c|k'_c = 1\}, \cdots, \{c|k'_c = K\})$

1: $\hat{\boldsymbol{U}}^{(t)} = \underset{\boldsymbol{U}\in\mathbb{R}^{Kd\times K}:\boldsymbol{U}^\top\boldsymbol{U}=I}{\operatorname{argmax}} \boldsymbol{U}^\top\big(\Sigma_{c=1}^C \boldsymbol{g}_c^{(t)}\boldsymbol{g}_c^{(t)\top}\big)\boldsymbol{U}$ ▷ Leading singular vectors

2: $(\hat{\boldsymbol{z}}^{(t+1)}, \{\bar{\boldsymbol{h}}_k^{(t+1)}\}_{k=1}^K) = \underset{\boldsymbol{z},\{\boldsymbol{h}_k\}_{k=1}^K}{\operatorname{argmin}} \sum_{c=1}^C \|(\hat{\boldsymbol{U}}^{(t)\top}\mathbf{G}^{(t)})_{:,c} - \boldsymbol{h}_{z_c}\|^2$ ▷ Spectral clustering

3: $\hat{\psi} = \underset{\psi\in\Psi}{\operatorname{argmax}}|\{c : \psi(\hat{z}_c^{(t+1)}) = k_c\}|$ ▷ $k_c = \{k|c \in \mathcal{S}_k\}$

4: Set $k'_c = \hat{\psi}(\hat{z}_c^{(t+1)})\ \forall c$ ▷ New model index assignment

B.1. Proof of Lemma 1

In this section, we establish an upper bound on the number of clients inaccurately clustered. Our analysis demonstrates that this upper bound is decreasing as a function of t, indicating that increasing t enhances the accuracy of clustering. Recall that CFL-GP selects one of the K models in a round-robin manner every P cycles for broadcast to clients, leading to an update of a part of the clients' feature vectors. For example, at time $t = 1 + (k-1)P$, the k-th block row of the gradient profile matrix $\mathbf{G}$ is updated if $t < T_{\text{cl}}$. Subsequent updates occur at $1 + (k-1)P + nKP$ for $n \in \mathbb{N} \cup \{0\}$ where $\mathbb{N}$ is the set of natural numbers. We can formally define the set of update times for the k-th block row of $\mathbf{G}$ up to time t as follows.

$$\mathcal{T}_k(t) = \{1 + (k-1)P + nKP | n \in \mathbb{N} \cup \{0\}, 1 + (k-1)P + nKP < t, t < T_{\text{cl}}\}. \tag{7}$$

For every $t' \in \mathcal{T}_k(t)$, it holds that $t' < t$. The k-th block of the gradient profile matrix $\mathbf{G}^{(t)}$ is updated $|\mathcal{T}_k(t)|$ times up to the commencement of time step t.

The algorithm updates the client feature vectors with a smoothing ratio $\beta_t = \frac{1}{\lfloor t/(KP)\rfloor+1}$ during each clustering phase. This approach integrates multi-step gradient information into clustering decisions and ensures cumulative average updates of the feature vectors. Given that the k-th block of the gradient profile matrix $\mathbf{G}^{(t-1)}$ aggregates the average of past gradients, the k-th feature block vector for client c at t can be expressed as follows.

$$\boldsymbol{g}_{c,k}^{(t-1)} = \sum_{t_k\in\mathcal{T}_k(t)} \frac{\nabla F(\mathcal{D}_{k_c^*}, \boldsymbol{\theta}_k^{(t_k)})}{|\mathcal{T}_k(t)|} + \sum_{t_k\in\mathcal{T}_k(t)} \frac{\mathbf{e}_c(\boldsymbol{\theta}_k^{(t_k)})}{|\mathcal{T}_k(t)|}. \tag{8}$$

Here we clarify the usage of the time (iteration) notation $\cdot^{(t)}$. It should be noted that $\mathbf{G}^{(t-1)}$ is employed for the formation of client clusters $\{\mathcal{S}_1^{(t)}, \ldots, \mathcal{S}_k^{(t)}\}$. Based on this clustering, the models $\{\boldsymbol{\theta}_1^{(t)}, \ldots, \boldsymbol{\theta}_K^{(t)}\}$ undergo updates. In other words, the clustering results at the beginning of t rely on the information in $\mathbf{G}^{(t-1)}$. In our analysis of clustering accuracy at time t-specifically at the commencement of the t-th iteration process, we shall consider the gradient profile matrix, which is updated up to time $t-1$ and utilized to create client clusters for t.

Using this notation, we can represent the gradient profile matrix at $t-1$ as $\mathbf{G}^{(t-1)} = \mathbf{A}^{(t-1)} + \mathbf{E}^{(t-1)}$ where $\mathbf{A}^{(t-1)}$ and

$\mathbf{E}^{(t-1)}$ are given as follows.

$$\mathbf{A}^{(t-1)} = \begin{pmatrix} \sum_{t_1 \in \mathcal{T}_1(t)} \frac{\nabla F(\mathcal{D}_{k_1^*}, \boldsymbol{\theta}_1^{(t_1)})}{|\mathcal{T}_1(t)|} & \cdots & \sum_{t_1 \in \mathcal{T}_1(t)} \frac{\nabla F(\mathcal{D}_{k_C^*}, \boldsymbol{\theta}_1^{(t_1)})}{|\mathcal{T}_1(t)|} \\ \vdots & \ddots & \vdots \\ \sum_{t_K \in \mathcal{T}_K(t)} \frac{\nabla F(\mathcal{D}_{k_1^*}, \boldsymbol{\theta}_K^{(t_K)})}{|\mathcal{T}_K(t)|} & \cdots & \sum_{t_K \in \mathcal{T}_K(t)} \frac{\nabla F(\mathcal{D}_{k_C^*}, \boldsymbol{\theta}_K^{(t_K)})}{|\mathcal{T}_K(t)|} \end{pmatrix} \tag{9}$$

$$\mathbf{E}^{(t-1)} = \begin{pmatrix} \sum_{t_1 \in \mathcal{T}_1(t)} \frac{\mathbf{e}_1(\boldsymbol{\theta}_1^{(t_1)})}{|\mathcal{T}_1(t)|} & \cdots & \sum_{t_1 \in \mathcal{T}_1(t)} \frac{\mathbf{e}_C(\boldsymbol{\theta}_1^{(t_1)})}{|\mathcal{T}_1(t)|} \\ \vdots & \ddots & \vdots \\ \sum_{t_K \in \mathcal{T}_K(t)} \frac{\mathbf{e}_1(\boldsymbol{\theta}_K^{(t_K)})}{|\mathcal{T}_K(t)|} & \cdots & \sum_{t_K \in \mathcal{T}_K(t)} \frac{\mathbf{e}_C(\boldsymbol{\theta}_K^{(t_K)})}{|\mathcal{T}_K(t)|} \end{pmatrix}. \tag{10}$$

Equations (9) and (10) illustrate that the gradient profile matrix $\mathbf{G}^{(t-1)}$ is a composite of the average population loss gradient matrix and the stochastic gradient noise (SGN) matrix, with varying row update frequencies at a given time t. We can decompose $\mathbf{G}^{(t-1)}$ into a central component, $\mathbf{A}^{(t-1)}$, and a noise component, $\mathbf{E}^{(t-1)}$. Based on this notion, the subsequent subsections will delve into spectral clustering using this matrix framework and analyze the iteration-wise clustering accuracy.

B.1.1. Spectral Clustering

To generate the clustering results used for the update of the models at time (t), the spectral clustering is applied to a realization of the random matrix $\mathbf{G}^{(t-1)}$. For ease of discussion, the superscript time notation $\cdot^{(t-1)}$ is omitted, presuming a constant and specified $t-1$. Therefore, $\mathbf{G}^{(t-1)}$, $\mathbf{A}^{(t-1)}$, and $\mathbf{E}^{(t-1)}$ are referred to as $\mathbf{G}$, $\mathbf{A}$, and $\mathbf{E}$. Additionally, in accordance with Algorithm 3, the estimated cluster identity $\hat{\boldsymbol{z}}^{(t)}$ for the c-th client, derived from $\{\boldsymbol{g}_c^{(t-1)}\}_{c=1}^C$, is simply denoted as $\hat{\boldsymbol{z}}$. In this subsection, we deal with the *realization of these random variables*.

Spectral clustering commences by identifying the top-K leading singular vectors of the gradient profile matrix $\mathbf{G}$. It should be noted that extraction of these singular vectors from $\mathbf{G}$ does not strictly mandate the use of Singular Value Decomposition (SVD). Nevertheless, for the sake of brevity within this proof, we utilize a reduced SVD approach to process the matrix $\mathbf{G}$, as follows.

$$\mathbf{G} = \sum_{i=1}^{C} \hat{\sigma}_i \hat{\boldsymbol{u}}_i \hat{\boldsymbol{v}}_i^\top \tag{11}$$

where $\hat{\sigma}_i$ represents the i-th largest singular value of $\mathbf{G}$, $\hat{\boldsymbol{u}}_i$ and $\hat{\boldsymbol{v}}_i$ are the left and right singular vectors corresponding to $\hat{\sigma}_i$, respectively. According to the definition of SVD, the matrix containing the top-K singular vectors can be represented as follows where $K < C$.

$$\hat{\boldsymbol{U}} = \underset{\boldsymbol{U} \in \mathbb{R}^{Kd \times K} : \boldsymbol{U}^\top \boldsymbol{U} = I}{\operatorname{argmax}} \boldsymbol{U}^\top \left(\Sigma_{c=1}^C \mathbf{G}_{:,c} \mathbf{G}_{:,c}^\top \right) \boldsymbol{U} = (\hat{\boldsymbol{u}}_1, ..., \hat{\boldsymbol{u}}_K). \tag{12}$$

After obtaining the matrix $\hat{\boldsymbol{U}}$, the dimensionality of the gradient profile matrix is reduced from $\mathbf{G} \in \mathbb{R}^{Kd \times C}$ to $\hat{\boldsymbol{U}}^\top \mathbf{G} \in \mathbb{R}^{K \times C}$. Next, the clustering is performed based on the low-dimensional matrix $\hat{\boldsymbol{U}}^\top \mathbf{G}$.

The objective of the clustering process is to acquire K cluster centers denoted as $\{\bar{\boldsymbol{h}}_k\}_{k=1}^K$, where $\bar{\boldsymbol{h}}_k \in \mathbb{R}^K$, and to estimate the cluster identities of the clients, denoted as $(\hat{z}_1, ..., \hat{z}_C) \in [K]^C$. The vectorized form of the cluster indices of the clients is represented by $\hat{\boldsymbol{z}} \in [K]^C$ where c-th element of $\hat{\boldsymbol{z}}$ is $\hat{z}_c$. The complete process of clustering, which involves dimensionality reduction and obtaining K cluster centers, is commonly known as spectral clustering and can be represented as follows.

$$(\hat{\boldsymbol{z}}, \{\bar{\boldsymbol{h}}_k\}_{k=1}^K) = \underset{\boldsymbol{z}, \{\boldsymbol{h}_k\}_{k=1}^K}{\operatorname{argmin}} \sum_{c=1}^{C} \|(\hat{\boldsymbol{U}}^\top \mathbf{G})_{:,c} - \boldsymbol{h}_{z_c}\|^2 \tag{13}$$

where $\boldsymbol{z}$ denotes the vectorized form of the cluster identities, and the c-th element of the vector $\boldsymbol{z} \in [K]^C$ is represented by $z_c \in [K]$. This clustering divides the clients into K disjoint sets.

It is important to note that $\hat{z}_c$ only represents the index of the cluster to which the c-th client is assigned, within the K distinct clusters generated by solving (13). $\hat{z}_c$ does not yield any information regarding the utilization of the model. The clusters of clients identified through (13) are mapped to corresponding models via an additional process, the cluster-model index matching process, as shown in Lines 3-4 in Algorithm 3. The specifics of this process, including a detailed explanation and its justification, are provided in Appendix B.2.2.

The ultimate goal of utilizing spectral clustering is to obtain the client clustering assignment $\hat{\boldsymbol{z}}$. It is worth noting that the clustering result, $\hat{\boldsymbol{z}}$, from Problem (13) can also be achieved by performing clustering on the rank-K approximated matrix of $\mathbf{G}$, which is denoted as $\hat{\mathbf{A}}$, along with index permutation. More specifically, we adopt the following lemma.

Lemma 2 (Application of Lemma 4.1 in (Löffler et al., 2021))**.** *Consider a K-means clustering problem on the rank-K matrix $\hat{\mathbf{A}}$ as follows.*

$$\hat{\mathbf{A}} = \sum_{i=1}^{K} \hat{\sigma}_i \hat{\boldsymbol{u}}_i \hat{\boldsymbol{v}}_i^\top \tag{14}$$

$$\left(\hat{\boldsymbol{z}}', \{\bar{\boldsymbol{a}}_k\}_{k=1}^K\right) = \underset{\boldsymbol{z}, \{\boldsymbol{a}_k\}_{k=1}^K}{\operatorname{argmin}} \sum_{c=1}^{C} \|\hat{\mathbf{A}}_{:,c} - \boldsymbol{a}_{z_c}\|^2. \tag{15}$$

Then there exists a permutation function $\psi \in \Psi$ such that $\hat{z}_c = \psi(\hat{z}'_c)\ \forall\ c$.

Lemma 2 states that the clustering results obtained from spectral clustering in Problem (13) can also be achieved by using the matrix $\hat{\mathbf{A}}$ which is the rank-K approximation of $\mathbf{G}$. Note that if the set $\{\hat{z}'_c\}_{c=1}^C$ can be transformed into $\{\hat{z}_c\}_{c=1}^C$ through a specific permutation function, then the clusters represented by the two sets are considered identical. Due to the equivalence of the clustering result, without loss of generality, we use $\hat{\boldsymbol{z}}'$ and $\{\bar{\boldsymbol{a}}_k\}_{k=1}^K$ instead of $\hat{\boldsymbol{z}}$ and $\{\bar{\boldsymbol{h}}_k\}_{k=1}^K$ for further analysis.

Now we use the proof technique of Lemma 4.2 in Löffler's work (Löffler et al., 2021) to derive (16)-(21) and rewrite the logic for readability.

We first establish the lower bound of the spectral norm of the SGN matrix $\|\mathbf{E}\|$. This is achieved through the application of the following.

$$\begin{aligned}\|\hat{\mathbf{A}} - \mathbf{A}\|_{\mathrm{F}} \leq \sqrt{2K}\|\hat{\mathbf{A}} - \mathbf{A} + \mathbf{G} - \mathbf{G}\| &\leq \sqrt{2K}(\|\hat{\mathbf{A}} - \mathbf{G}\| + \|\mathbf{A} - \mathbf{G}\|) \\ &\leq 2\sqrt{2K}\|\mathbf{A} - \mathbf{G}\| = 2\sqrt{2K}\|\mathbf{E}\|.\end{aligned} \tag{16}$$

The first inequality in (16) can be obtained from the inequality of the Frobenius norm and the spectral norm. The second inequality is satisfied by the triangle inequality of the spectral norm. As we denote in (14), $\hat{\mathbf{A}}$ is an optimal rank-K approximation of the matrix $\mathbf{G}$ with respect to the Frobenius norm and spectral norm by the properties of SVD. Based on this, the third inequality is satisfied.

Now Consider a matrix $\bar{\mathbf{A}} \in \mathbb{R}^{Kd \times C}$ whose c-th column is $\bar{\boldsymbol{a}}_{\hat{z}'_c}$ as $\bar{\mathbf{A}} = (\bar{\boldsymbol{a}}_{\hat{z}'_1}, ..., \bar{\boldsymbol{a}}_{\hat{z}'_C})$. Then $\|\hat{\mathbf{A}} - \mathbf{A}\|_{\mathrm{F}}$ is lower-bounded as follows.

$$\frac{1}{2}\|\bar{\mathbf{A}} - \mathbf{A}\|_{\mathrm{F}} \leq \frac{1}{2}\|\bar{\mathbf{A}} - \hat{\mathbf{A}}\|_{\mathrm{F}} + \frac{1}{2}\|\mathbf{A} - \hat{\mathbf{A}}\|_{\mathrm{F}} \leq \|\hat{\mathbf{A}} - \mathbf{A}\|_{\mathrm{F}}. \tag{17}$$

The first inequality of (17) holds by the triangle inequality, and the second inequality holds by the definition of $\bar{\mathbf{A}}$ in (15).

Consider a set $\mathcal{W} = \{c : \|\bar{\boldsymbol{a}}_{\hat{z}'_c} - \mathbf{A}_{:,c}\| \geq \Delta_g/2\}$, which is a set of clients where the corresponding clients show the l_2-norm of the gap between the estimated cluster center for them and the corresponding concatenated population gradient is greater than $\Delta_g/2$. Then, the cardinality of the set $\mathcal{W}$ is bounded as follows.

$$|\mathcal{W}| \leq \frac{\|\bar{\mathbf{A}} - \mathbf{A}\|_{\mathrm{F}}^2}{(\Delta_g/2)^2} \leq \frac{128K\|\mathbf{E}\|^2}{\Delta_g^2}. \tag{18}$$

Now we state that all the clients belonging to the complement of $\mathcal{W}$ are correctly clustered. Consequently, the number of clients clustered erroneously is bounded above by the cardinality of $\mathcal{W}$. Consider sets $\{\tilde{\mathcal{Y}}_k\}_{k=1}^K$ which are defined as

$\tilde{\mathcal{Y}}_k = \{c \in [C] : k_c^* = k, c \in (\mathcal{W})^{\complement}\}, \forall k$ where $(\mathcal{W})^{\complement}$ is the complement of the set $\mathcal{W}$, and $\tilde{\mathcal{Y}}_k$ is non-empty. Note that the *correctly clustered clients* c and c' means that if the cluster indices from the output of the clustering are different, then the true distribution identities of the clients c and c' are also different. The statement is equivalent to the following.

$$\nexists c, c' \in [C] : c \in \tilde{\mathcal{Y}}_k, c' \in \tilde{\mathcal{Y}}_{k'} \text{ and } \hat{z}'_c = \hat{z}'_{c'} \tag{19}$$

where $k, k' \in [K], k \neq k'$.

To prove the statement (19), we employ a proof by contradiction. Let us assume that the statement is false. Then there exist $c \in \tilde{\mathcal{Y}}_k$ and $c' \in \tilde{\mathcal{Y}}_{k'}$ such that $\hat{z}'_c = \hat{z}'_{c'}$. This implies that the spectral clustering algorithm has grouped clients c and c' together into the same cluster, even though their true distribution identities are different and both clients belong to the complement of $\mathcal{W}$. By the definition of Δ_g, we have

$$\Delta_g \leq \left\| \begin{pmatrix} \sum_{t_1 \in \mathcal{T}_1(t)} \frac{\nabla F(\mathcal{D}_{k_c^*}, \boldsymbol{\theta}_1^{(t_1)})}{|\mathcal{T}_1(t)|} \\ \vdots \\ \sum_{t_K \in \mathcal{T}_K(t)} \frac{\nabla F(\mathcal{D}_{k_c^*}, \boldsymbol{\theta}_K^{(t_K)})}{|\mathcal{T}_K(t)|} \end{pmatrix} - \begin{pmatrix} \sum_{t_1 \in \mathcal{T}_1(t)} \frac{\nabla F(\mathcal{D}_{k_{c'}^*}, \boldsymbol{\theta}_1^{(t_1)})}{|\mathcal{T}_1(t)|} \\ \vdots \\ \sum_{t_K \in \mathcal{T}_K(t)} \frac{\nabla F(\mathcal{D}_{k_{c'}^*}, \boldsymbol{\theta}_K^{(t_K)})}{|\mathcal{T}_K(t)|} \end{pmatrix} \right\|. \tag{20}$$

Note that the right-hand side of the inequality (20) is equal to $\|\mathbf{A}_{:c} - \mathbf{A}_{:c'}\|$. By the triangle inequality of the norm, we have an upper bound for $\|\mathbf{A}_{:c} - \mathbf{A}_{:c'}\|$ as follows.

$$\|\mathbf{A}_{:c} - \bar{\boldsymbol{a}}_{\hat{z}'_c}\| + \left\|\bar{\boldsymbol{a}}_{\hat{z}'_c} - \bar{\boldsymbol{a}}_{\hat{z}'_{c'}}\right\| + \left\|\mathbf{A}_{:c'} - \bar{\boldsymbol{a}}_{\hat{z}'_{c'}}\right\| < \Delta_g \tag{21}$$

where the inequality holds due to the assumption of $\hat{z}'_c = \hat{z}'_{c'}$ and the definition of $\mathcal{W}$ as c and c' are clients belonging to the complement of set $\mathcal{W}$.

It is straightforward to see that the inequalities (20) and (21) are contradictory to the definition of Δ_g since the right-hand side term of the inequality (20) is smaller than or equal to the left-hand side of (21).

B.1.2. Upper bound of the number of incorrectly clustered clients

Up to this point, we have established an upper bound for the number of incorrectly clustered clients as in (18), which is shown to be directly proportional to the square of the spectral norm of the SGN matrix. Given that the SGN vectors exhibit Gaussian characteristics, the spectral norm of the matrix $\mathbf{E}^{(t-1)}$ can be upper-bounded with high probability. It is noteworthy that as t increases, the spectral norm of the matrix tends to decrease. This is attributed to the fact that the act of averaging a larger number of independent-centered Gaussian vectors is more likely to yield an SGN matrix with reduced variance.

Recall the each block row of matrix $\mathbf{G}$ has been updated at least T_a times, where T_a is defined as $T_a = \lfloor \frac{t-1}{KP} \rfloor$. The clustering results observable at the beginning of the t-th iteration are obtained by the $t-1$-th gradient profile matrix. The $t-1$-th gradient profile matrix $\mathbf{G}$ is updated $\lfloor (t-1)/P \rfloor$ times, given that P denotes the clustering period. Since the K different blocks of $\mathbf{G}$ are updated in a round-robin manner, each block is updated at least $\lfloor \frac{t-1}{KP} \rfloor$ times up to t. Therefore, the set $|\mathcal{T}_K(t)|$ contains more than or equal to T_a elements and we have $|\mathcal{T}_K(t)| \geq T_a$ for a given t.

Based on this observation and the given assumptions for Lemma 1, we introduce the following lemma for the SGN matrix $\mathbf{E}^{(t-1)}$.

Lemma 3. *Suppose that Assumptions 1, 2, 3 hold. Consider any $\delta \in (0, 1)$ and recall the representation of the gradient profile matrix $\mathbf{G}^{(t-1)}$, $\mathbf{G}^{(t-1)} = \mathbf{A}^{(t-1)} + \mathbf{E}^{(t-1)}$ where $\mathbf{A}^{(t-1)}$ and $\mathbf{E}^{(t-1)}$ are given as (9). The spectral norm of $\mathbf{E}^{(t-1)}$ is bounded as $\|\mathbf{E}^{(t-1)}\| \lesssim \frac{C}{\delta} \left(\frac{v^2}{T_a b}\right)^{\frac{1}{2}}$ with probability at least $1 - \delta$.*

Proof. See Appendix B.3. □

Lemma 3 states that, with high probability, the spectral norm of the SGN matrix is upper-bounded subsequent to the completion of $t-1$ iterations.

Finally, combining (18) and Lemma 3, we have the following inequalities which hold with probability at least $1-\delta$.

$$|\mathcal{W}| \leq \frac{128K\|\mathbf{E}^{(t-1)}\|^2}{\Delta_g^2} \lesssim \frac{KC^2v^2}{\Delta_g^2\delta^2 T_a b}. \tag{22}$$

This observation directly implies that the upper bound on the number of incorrectly clustered clients can be characterized by the order of $\mathcal{O}(\frac{1}{\delta^2 t})$.

B.2. Proof of Theorem 1

Proof Sketch. The proof of Theorem 1 is established by sequentially addressing the following problems.

- [**Spectral clustering**] In B.2.1, we first explore the accuracy of spectral clustering based on the gradient profile matrix for client clustering. We employ Lemma 1 to compute the count of clients that are incorrectly clustered during the training process.
- [**Cluster-model index matching**] Secondly, in B.2.2, we delve into the cluster-model index matching, which focuses on assigning the client sets obtained from spectral clustering to their respective models. The main objective of this process is to ensure that the majority of clients are consecutively assigned to a particular model for updating. This approach guarantees that each model is ultimately updated to an optimal model for the corresponding distribution. We will show that by correctly clustering a sufficient number of clients and employing an appropriate permutation function, the majority of clients with a specific distribution identity can consistently contribute to updating the same model during the training process.
- [**Cluster driven model update**] Finally, in B.2.3, we shift our focus to the model updates based on the clustering outcomes from the preceding steps. Within each cluster, the central unit collects gradients from the associated clients and performs a model update on the corresponding model. We establish the contractive property of the algorithm under the potential presence of the incorrectly clustered clients within each cluster.

B.2.1. Spectral Clustering

To estimate the number of clients incorrectly clustered at time t, we utilize Lemma 3, particularly under the conditions outlined in Theorem 1. We present Proposition 2 to formalize this.

Proposition 2. *Recall that $|W(t)|$ represents the number of incorrectly clustered clients at t. For any given $\delta \in (0,1)$, it can be shown that with a probability of at least $1-\frac{\delta}{6}$, $|W(t)| < \frac{C\rho}{4}$.*

The proof of this proposition commences by invoking the established bound on the spectral norm of $\mathbf{E}^{(t-1)}$ from Lemma 3. This lemma indicates that $\|\mathbf{E}^{(t-1)}\| \leq \frac{\zeta' C}{\delta}\left(\frac{v^2}{T_a b}\right)^{\frac{1}{2}}$ with probability at least $1-\delta$. In the same context of Lemma 3, the term T_a is defined as the minimum number of updates that any k-th block has received. This definition implies that each block row of the gradient profile matrix $\mathbf{G}^{(t-1)}$ has undergone updates at least T_a times. Based on the assumption, we have $T_a = \lfloor \frac{t-1}{KP} \rfloor = \lfloor (\frac{\lambda K^2 PCv^2}{\delta^2\Delta_g^2\rho b})/(KP)+1 \rfloor$. Here, λ is defined as $\lambda = (96\sqrt{2}\zeta')^2+1$ with ζ' being the universal constant outlined in Lemma 3 and λ is not problem-specific but is instead a universal constant.

Consequently, with probability at least $1-\frac{\delta}{6}$, we have $\|\mathbf{E}^{(t-1)}\| \leq \frac{6\zeta' C}{\delta}\left(\frac{v^2}{T_a b}\right)^{\frac{1}{2}}$. Further, incorporating the condition $T_a \geq \frac{\lambda KCv^2}{\delta^2\Delta_g^2\rho b}$ into the inequality (22), we obtain

$$|\mathcal{W}^{(t)}| \leq \frac{128K\|\mathbf{E}^{(t-1)}\|^2}{\Delta_g^2} \leq \frac{128K(\zeta' Cv)^2}{\Delta_g^2(\delta/6)^2 T_a b} < \frac{C\rho}{4}. \tag{23}$$

In other words, at time t, the number of incorrectly clustered clients is upper bounded by $|W(t)| < \frac{C\rho}{4}$ with probability at least $1-\frac{\delta}{6}$. Furthermore, it is fact that $|\mathcal{W}^{(t+1)}| < \frac{C\rho}{4}$ also holds with probability at least $1-\frac{\delta}{6}$. An event of interest is defined wherein both conditions $|W(t)| < \frac{C\rho}{4}$ and $|W(t+1)| < \frac{C\rho}{4}$ are concurrently satisfied as $\mathcal{E}_1 = \left\{|W(t)| < \frac{C\rho}{4}, |W(t+1)| < \frac{C\rho}{4}\right\}$. $\mathcal{E}_1$ holds with probability at least $1-\frac{\delta}{3}$. Hereafter, we assume that $\mathcal{E}_1$ holds.

B.2.2. CLUSTER-MODEL INDEX MATCHING

Up until now, we have established the upper bound for the number of incorrectly clustered clients based on the results obtained from the spectral clustering. However, clustering itself does not provide information on which model each client should participate in. Therefore, in our algorithm, we compare the clustering results obtained from consecutive steps and match each of the K groups, the clustered clients and the models, one-to-one. Among several one-to-one matching methods, we select a function that can preserve the cluster identity for the majority of clients and permute $\hat{\boldsymbol{z}}'$ obtained from the current spectral clustering accordingly. This subsection will focus on this *Cluster-Model Index Matching* that determines which model updates each client should participate in using the results of spectral clustering.

Consider the K models at t, $\boldsymbol{\theta}_1^{(t)}, ..., \boldsymbol{\theta}_K^{(t)}$ and recall the definition of the *client-plurality model* for clients with distribution $\mathcal{D}_k$ as $\hat{\boldsymbol{\theta}}_k^{(t)} = \boldsymbol{\theta}_{M(k,t)}^{(t)}$ where $M(k,t) \in \arg\max_{m\in[K]} |\mathcal{S}_k^* \cap \mathcal{S}_m^{(t)}|$. The function $M(k,t)$ designates the model index most frequently chosen by the plurality of clients with distribution identity k at time t. Similarly, we can consider $M(k,t+1)$, which represents the model index most frequently chosen by the plurality of clients with distribution identity k at time $t+1$. The objectives of this subsection B.2.2 are twofold. First, we aim to demonstrate that after updating the models resulting in $\boldsymbol{\theta}_1^{(t+1)}, \ldots, \boldsymbol{\theta}_K^{(t+1)}$, the K client-plurality models are distinct. In other words, each set of clients associated with a particular data distribution selects different models as their respective client-plurality models. Secondly, we aim to demonstrate that $M(k,t) = M(k,t+1)$, thereby indicating that the associations between the model indices and the distribution indices remain unchanged from time step t to $t+1$ with high probability.

We begin by introducing the following proposition.

Proposition 3. *Consider any $k \in [K]$ and $m = M(k,t)$. Then, we have*

$$|\mathcal{S}_k^* \cap \mathcal{S}_m^{(t)}| \geq 0.75C\rho. \tag{24}$$

Proposition 3 indicates that more than $0.75C\rho$ clients who have k as its distribution index have been grouped together into the given set $\mathcal{S}_m^{(t)}$. It can be shown by contradiction. Assume that the given statement does not hold. Then, there exists $k \in [K]$ such that $|\mathcal{S}_k^* \cap \mathcal{S}_m^{(t)}| < 0.75C\rho$. It indicates that more than $\min_k(|\mathcal{S}_k^*|) - 0.75C\rho$ clients are incorrectly clustered and implies $W(t) \geq 0.25C\rho$. Such a result is in direct contradiction to the assumption that event $\mathcal{E}_1$ is true. We also have the following proposition.

Proposition 4. *Consider any two distinct distribution indices k_1 and k_2 such that $k_1, k_2 \in [K]$ and $k_1 \neq k_2$. Then, it holds that $M(k_1,t) \neq M(k_2,t)$.*

In simpler terms, for any two distinct distribution indices k_1 and k_2, the plurality of clients associated with k_1 are mapped to the model $M(k_1,t)$, and those associated with k_2 are mapped to $M(k_2,t)$. Then, $M(k_1,t)$ and $M(k_2,t)$ are distinct. Proposition 4 can also be established through a proof by contradiction. Assume the existence of distribution indices k_1, k_2 such that $M(k_1,t) = M(k_2,t) = m$ where $k_1 \neq k_2$, and $k_1, k_2 \in [K]$. Then, according to Proposition 3, in the client set $\mathcal{S}_m^{(t)}$, we have that $|\mathcal{S}_{k_1}^* \cap \mathcal{S}_m^{(t)}| \geq 0.75C\rho$. and $|\mathcal{S}_{k_2}^* \cap \mathcal{S}_m^{(t)}| \geq 0.75C\rho$. It directly implies that $W(t) \geq 0.75C\rho$, which contradicts the underlying assumption that event $\mathcal{E}_1$ holds.

Now, we analyze a non-trivial case where the algorithm performs a cluster update, potentially altering client model indices. For the trivial scenario where no cluster update occurs, as indicated by line 12 of Algorithm 2, we omit the proof since the condition $M(k,t) = M(k,t+1)$ for all k naturally holds, assuming stable client assignments across iterations.

Recall the spectral clustering subroutine in Algorithm 3, which generates sets of clients $\{\mathcal{Q}_z^{(t+1)}\}_{z=1}^K$ where

$$\mathcal{Q}_z^{(t+1)} = \{c | \hat{z}'^{(t+1)}_c = z\}. \tag{25}$$

In other words, the set $\mathcal{Q}_z^{(t+1)}$ consists of clients whose cluster identity is designated as z. As previously discussed in B.1, the clients in the set $\mathcal{Q}_z^{(t+1)}$ do not necessarily participate in the update for the z-th model or z-th distribution $\mathcal{D}_z$. For all $z \in [K]$, $\mathcal{Q}_z^{(t+1)}$ represents the collection of clients assigned to the z-th group, and no permutation or cluster-index matching has been applied to the clusters.

For the clients sets $\{\mathcal{Q}_1^{(t+1)}, \ldots, \mathcal{Q}_K^{(t+1)}\}$, consider a mapping $\tilde{M}(\cdot,\cdot) : [K] \times \mathbb{N} \mapsto [K]$ such that for a given k and t $\tilde{M}(k,t) \in \arg\max_{m\in[K]} |\mathcal{S}_k^* \cap \mathcal{Q}_m^{(t)}|$. Consequently, the characteristics of $\{\mathcal{S}_1^{(t)}, \ldots \mathcal{S}_K^{(t)}\}$ as described in Propositions 3 and 4 hold for $\{\mathcal{Q}_1^{(t+1)}, \ldots, \mathcal{Q}_K^{(t+1)}\}$ as follows.

Proposition 5. *Consider any $k \in [K]$, t, and $z = \tilde{M}(k,t)$. Then, $|\mathcal{S}_k^* \cap \mathcal{Q}_z^{(t+1)}| \geq 0.75C\rho$. In addition, consider any two distinct distribution indices k_1 and k_2 such that $k_1, k_2 \in [K]$ and $k_1 \neq k_2$. Then, it holds that $\tilde{M}(k_1,t) \neq \tilde{M}(k_2,t)$.*

Now, our focus shifts to the permutation $\hat{\psi}$ utilized in Algorithm 3. This permutation function establishes a one-to-one mapping between the K disjoint clusters, $\{\mathcal{Q}_z^{(t+1)}\}_{z=1}^K$ and $\{\mathcal{S}_m^{(t)}\}_{m=1}^K$. Notably, it aims to maximize the number of clients who retain their assigned model index. We argue that the permutation function $\hat{\psi}$ can be obtained through a process in which each client cluster $\mathcal{Q}_z^{(t+1)}$ selects a cluster from $\{\mathcal{S}_m^{(t)}\}_{m=1}^K$, in a greedy fashion, opting for the one with the most significant client intersection. This argument is elaborated in the subsequent Lemma 4.

Lemma 4. *For a given client $c \in [C]$, the model index obtained by the permutation function $\hat{\psi}$, $k_c^{(t+1)} = \hat{\psi}(\hat{z}'^{(t+1)}_c)$, is equal to $\arg\max_{m\in[K]} |\mathcal{S}_m^{(t)} \cap \{c' \in [C] | \hat{z}'^{(t+1)}_{c'} = \hat{z}'^{(t+1)}_c\}|$.*

Proof. Given any two distinct model indices m_1 and m_2 such that $m_1 \neq m_2$, and corresponding sets of clients $\mathcal{S}_{m_1}^{(t)}$ and $\mathcal{S}_{m_2}^{(t)}$, we define z_1 and z_2 as follows.

$$z_1 = \arg\max_{z\in[K]} |\mathcal{S}_{m_1}^{(t)} \cap \mathcal{Q}_z^{(t+1)}|, z_2 = \arg\max_{z\in[K]} |\mathcal{S}_{m_2}^{(t)} \cap \mathcal{Q}_z^{(t+1)}|. \tag{26}$$

We assert that

$$z_1 \neq z_2. \tag{27}$$

Moreover, we have

$$m_1 = \arg\max_{m\in[K]} |\mathcal{S}_m^{(t)} \cap \mathcal{Q}_{z_1}^{(t+1)}|, m_2 = \arg\max_{m\in[K]} |\mathcal{S}_m^{(t)} \cap \mathcal{Q}_{z_2}^{(t+1)}|. \tag{28}$$

We note that this fact completes the proof since there exists a one-to-one mapping between $\{\mathcal{S}_m^{(t)}\}_{m=1}^K$ and $\{\mathcal{Q}_z^{(t+1)}\}_{z=1}^K$ with the maximum intersection operation.

To validate the assertion (27), we refer to Propositions 3 and 5, implying that for each $m \in [K]$, there exists a unique distribution index k such that $M(k,t) = m$ and a corresponding cluster index z^* such that $\tilde{M}(k,t+1) = z^*$. This leads us to the following proposition.

Proposition 6. *For a given model index $m \in [K]$ and corresponding distribution index k such that $M(k,t) = m$, consider a unique cluster index $z^* = \tilde{M}(k,t+1)$. Then, it holds that*

$$z^* = \arg\max_{z\in[K]} |\mathcal{S}_m^{(t)} \cap \mathcal{Q}_z^{(t+1)}|. \tag{29}$$

Proof of Proposition 6 is straightforward as follows. Recall that $\mathcal{S}_m^{(t)}$ comprises more than $0.75C\rho$ clients with distribution identity k and fewer than $0.25C\rho$ clients with other distribution identities. Similarly, $\mathcal{Q}_{z^*}^{(t+1)}$ contains more than $0.75C\rho$ clients of identity k. The intersection of these client sets must exceed $0.5C\rho$. If $\mathcal{Q}_{z^*}^{(t+1)}$ had fewer than $0.5C\rho$ clients of identity k from $\mathcal{S}_m^{(t)}$, it would imply a misallocation exceeding $0.25C\rho$ clients, which contradicts to the underlying assumption that event $\mathcal{E}_1$, indicating the number of incorrectly clustered clients is less than $0.25C\rho$, holds.

Note that for any other cluster index $\bar{z} \in [K]$ such that $\bar{z} \neq z^*$, $|\mathcal{S}_m^{(t)} \cap \mathcal{Q}_{\bar{z}}^{(t+1)}| \leq 0.25C\rho$ which completes the proof of Proposition 6.

Further, consider the two distinct distribution indices k_1 and k_2 such that $M(k_1,t) = m_1$ and $M(k_2,t) = m_2$. Thus, $z_1 = \tilde{M}(k_1,t+1)$ and $z_2 = \tilde{M}(k_2,t+1)$, confirming $z_1 \neq z_2$ and completing the proof. Similarly, as shown in Proposition 6, Equation (28) holds trivially. □

Lemma 4 suggests that the result of establishing a one-to-one correspondence between newly formed and prior client clusters can also be achieved by a greedy matching strategy. This strategy involves each $\{\mathcal{Q}_z^{(t+1)}\}_{z=1}^K$ selecting one of $\{\mathcal{S}_m^{(t)}\}_{m=1}^K$ which it has the most significant overlap. In the c-th client's perspective, the new model index will be given as $k_c^{(t+1)} = \arg\max_{m\in[K]} |\mathcal{S}_m^{(t)} \cap \{c' \in [C] | \hat{z}'^{(t+1)}_{c'} = \hat{z}'^{(t+1)}_c\}|$.

Finally, based on Lemma 4, we have

$$\mathcal{S}_m^{(t+1)} = \underset{\{\mathcal{Q}_z^{(t+1)}\}_{z=1}^K}{\arg\max} \; |\mathcal{S}_m^{(t)} \cap \mathcal{Q}_z^{(t+1)}|, \forall m \in [K]. \tag{30}$$

It also indicates that for any distribution index $k \in [K]$, $M(k,t) = M(k,t+1)$ (See Proof of Lemma 4). Consequently, this denotes that clients characterized by the distribution index k are highly likely to adhere to the same model $M(k,t)$ across the consecutive steps, thereby indicating the stability of the client-model ties.

B.2.3. Cluster-driven Model update

In this subsection, we demonstrate that model convergence towards the optimal models for the respective K distributions occurs, with the error rate dependent on specific system parameters. Specifically, we show the contractive property of the gap between the *client-plurality models* and their corresponding optimal models. Our goal is to illustrate that the model chosen by the majority of clients associated with a particular distribution index k is updated towards its respective optimal model $\boldsymbol{\theta}_k^*$.

For a given model index $m \in [K]$ and its associated client set $\mathcal{S}_m^{(t)}$, there exists a unique distribution index k such that $M(k,t) = m$. To enhance notation clarity, we introduce $\hat{\mathcal{S}}_k^{(t)}$ to represent the set of clients whose plurality distribution index is k, satisfying $\hat{\mathcal{S}}_k^{(t)} = \mathcal{S}_{M(k,t)}^{(t)}$.

Our analysis focuses on the contractive property of the k-th client plurality model $\hat{\boldsymbol{\theta}}_k^{(t)}$ associated with $\hat{\mathcal{S}}_k^{(t)}$. CFL-GP updates the model based on the gradients from the corresponding client set $\hat{\mathcal{S}}_k^{(t)}$. After the update, the plurality of clients associated with the k-th distribution is assigned to the same model with high probability, as shown in the previous subsections. Based on this, we analyze the gap between the k-th client plurality model and the optimal model after the update.

We adopt the proof structure and methodology from Lemma 3 in (Ghosh et al., 2020), which analyzes the gradient update for a given model by decomposing it into contributions from correctly clustered clients and those from incorrectly clustered clients. The logic of the proof has been rephrased for clarity and readability. For the first step, we extend the term of the gradient update (31) as follows.

$$\|\hat{\boldsymbol{\theta}}_k^{(t+1)} - \boldsymbol{\theta}_k^*\| = \left\| \hat{\boldsymbol{\theta}}_k^{(t)} - \boldsymbol{\theta}_k^* - \frac{\alpha}{C} \sum_{c \in \hat{\mathcal{S}}_k^{(t)}} \nabla f(\mathcal{X}_c^{(t)}, \hat{\boldsymbol{\theta}}_k^{(t)}) \right\| \tag{31}$$

$$\leq \left\| \hat{\boldsymbol{\theta}}_k^{(t)} - \boldsymbol{\theta}_k^* - \frac{\alpha}{C} \sum_{c \in \mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}} \left(\nabla f(\mathcal{X}_c^{(t)}, \hat{\boldsymbol{\theta}}_k^{(t)}) + \nabla F(\mathcal{D}_{k_c^*}, \hat{\boldsymbol{\theta}}_k^{(t)}) - \nabla F(\mathcal{D}_{k_c^*}, \hat{\boldsymbol{\theta}}_k^{(t)}) \right) \right\|$$

$$+ \left\| \frac{\alpha}{C} \sum_{c \in \bar{\mathcal{S}}_k^* \cap \hat{\mathcal{S}}_k^{(t)}} \nabla f(\mathcal{X}_c^{(t)}, \hat{\boldsymbol{\theta}}_k^{(t)}) \right\| \tag{32}$$

where $\gamma = \alpha/C$ is a learning rate for the gradient descent update and $\bar{\mathcal{S}}_k^*$ is a set of clients whose true cluster identity is not k. We omit the notation t and the model index k for γ as in this analysis, as $\gamma_k^{(t)}$ is fixed for all t and k. The relation in equation (31) arises directly from the definition of the CFL-GP update mechanism. The subset of clients identified by their plurality distribution index of k contributes to the update of a corresponding model, which we refer to as the client plurality model for distribution index k, denoted by $\hat{\boldsymbol{\theta}}_k^{(t)}$. Consequently, the model updated at step t persists as the client plurality model for the subsequent step $t+1$, adhering to distribution k's client plurality model as $\hat{\boldsymbol{\theta}}_k^{(t+1)}$. This continuity is assured with high probability under the assumptions given for Theorem 1, as substantiated by the proofs presented in the preceding section.

Using the triangle inequality of the norm, we have

$$\|\hat{\boldsymbol{\theta}}_k^{(t+1)} - \boldsymbol{\theta}_k^*\| \leq \left\| \hat{\boldsymbol{\theta}}_k^{(t)} - \boldsymbol{\theta}_k^* - \frac{|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|\alpha}{C} \nabla F(\mathcal{D}_{k_c^*}, \hat{\boldsymbol{\theta}}_k^{(t)}) \right\| \tag{33}$$

$$+ \frac{|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|\alpha}{C} \left\| \nabla F(\mathcal{D}_{k_c^*}, \hat{\boldsymbol{\theta}}_k^{(t)}) - \frac{1}{|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|} \sum_{c \in \mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}} \nabla f(\mathcal{X}_c^{(t)}, \hat{\boldsymbol{\theta}}_k^{(t)}) \right\| \tag{34}$$

$$+ \left\| \frac{\alpha}{C} \sum_{c \in \bar{\mathcal{S}}_k^* \cap \hat{\mathcal{S}}_k^{(t)}} \nabla f(\mathcal{X}_c^{(t)}, \hat{\boldsymbol{\theta}}_k^{(t)}) \right\|. \tag{35}$$

Leveraging the properties of L-smoothness and the μ-strong convexity, we first establish an upper bound for the term on the right-hand side of inequality (33) as follows.

$$\left\| \hat{\boldsymbol{\theta}}_k^{(t)} - \boldsymbol{\theta}_k^* - \frac{|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|\alpha}{C} \nabla F(\mathcal{D}_{k_c^*}, \hat{\boldsymbol{\theta}}_k^{(t)}) \right\|^2 \tag{36}$$

$$= \|\hat{\boldsymbol{\theta}}_k^{(t)} - \boldsymbol{\theta}_k^*\|^2 - 2\frac{|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|\alpha}{C} \langle \hat{\boldsymbol{\theta}}_k^{(t)} - \boldsymbol{\theta}_k^*, \nabla F(\mathcal{D}_{k_c^*}, \hat{\boldsymbol{\theta}}_k^{(t)}) \rangle + \left(\frac{|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|\alpha}{C} \right)^2 \|\nabla F(\mathcal{D}_{k_c^*}, \hat{\boldsymbol{\theta}}_k^{(t)})\|^2 \tag{37}$$

$$\leq \left(1 - 2\frac{|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|\alpha}{C} \frac{\mu L}{\mu + L} \right) \|\hat{\boldsymbol{\theta}}_k^{(t)} - \boldsymbol{\theta}_k^*\|^2 + \frac{|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|\alpha}{C} \left(\frac{|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|\alpha}{C} - \frac{2}{\mu + L} \right) \|\nabla F(\mathcal{D}_{k_c^*}, \hat{\boldsymbol{\theta}}_k^{(t)})\|^2. \tag{38}$$

By selecting $\alpha = \frac{2}{\mu+L}$, we have $\frac{|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|\alpha}{C} < \frac{2}{\mu+L}$ and

$$\begin{aligned} \left\| \hat{\boldsymbol{\theta}}_k^{(t)} - \boldsymbol{\theta}_k^* - \frac{|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|\alpha}{C} \nabla F(\mathcal{D}_{k_c^*}, \hat{\boldsymbol{\theta}}_k^{(t)}) \right\|^2 &\leq \left(1 - 4\frac{|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|\mu L}{C(\mu + L)^2} \right) \|\hat{\boldsymbol{\theta}}_k^{(t)} - \boldsymbol{\theta}_k^*\|^2 \\ &\leq \left(1 - \frac{3\rho\mu L}{(\mu + L)^2} \right) \|\hat{\boldsymbol{\theta}}_k^{(t)} - \boldsymbol{\theta}_k^*\|^2 \end{aligned} \tag{39}$$

where the last inequality of (39) holds by (24) which indicates that for a given distribution index k, at least $0.75C\rho$ clients are correctly clustered.

Next, our objective is to establish an upper bound for the expectation of (34). It is important to note that all $\mathbf{x}$ instances within $\mathcal{X}_c^{(t)}$ that satisfy $c \in \mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}$ are data samples drawn from the distribution $\mathcal{D}_k$. As a result, owing to the independence among the minibatch samples and the bounded variance assumptions, we have the following.

$$\mathbb{E} \left\| \nabla F(\mathcal{D}_{k_c^*}, \hat{\boldsymbol{\theta}}_k^{(t)}) - \frac{1}{|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|} \sum_{c \in \mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}} \nabla f(\mathcal{X}_c^{(t)}, \hat{\boldsymbol{\theta}}_k^{(t)}) \right\|^2 \leq \frac{v^2}{b|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|}. \tag{40}$$

Now, consider any δ_1 such that $\delta_1 \in (0, 1)$. By applying Lemma 1, the Markov inequality, and taking into account that $|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}| \geq C\rho - \frac{128K\|\mathbf{E}^{(t-1)}\|^2}{\Delta_g^2}$, we can establish the following inequality with probability at least $1 - \delta_1$.

$$\frac{|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|\alpha}{C} \left\| \nabla F(\mathcal{D}_{k_c^*}, \hat{\boldsymbol{\theta}}_k^{(t)}) - \frac{1}{|\mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}|} \sum_{c \in \mathcal{S}_k^* \cap \hat{\mathcal{S}}_k^{(t)}} \nabla f(\mathcal{X}_c^{(t)}, \hat{\boldsymbol{\theta}}_k^{(t)}) \right\| \tag{41}$$

$$\leq \frac{2v}{\delta_1(\mu + L)\sqrt{b(C\rho - \frac{128K\|\mathbf{E}^{(t-1)}\|^2}{\Delta_g^2})}}. \tag{42}$$

As a next step, we aim to derive an upper bound for (35), $\|\frac{\alpha}{C}\sum_{c\in\bar{\mathcal{S}}_k^*\cap\hat{\mathcal{S}}_k^{(t)}}\nabla f(\mathcal{X}_c^{(t)},\hat{\boldsymbol{\theta}}_k^{(t)})\|$. It is important to note that $\sum_{c\in\bar{\mathcal{S}}_k^*\cap\hat{\mathcal{S}}_k^{(t)}}\nabla f(\mathcal{X}_c^{(t)},\hat{\boldsymbol{\theta}}_k^{(t)})$ is a sum of gradients with respect to $\hat{\boldsymbol{\theta}}_k^{(t)}$ where $\mathcal{X}_c^{(t)}$ comprises multiple data samples that do not adhere to the distribution $\mathcal{D}_k$. We can then proceed as follows.

$$\frac{\alpha}{C}\sum_{c\in\bar{\mathcal{S}}_k^*\cap\hat{\mathcal{S}}_k^{(t)}}\nabla f(\mathcal{X}_c^{(t)},\hat{\boldsymbol{\theta}}_k^{(t)})=\frac{\alpha}{C}\sum_{k'\neq k}\sum_{c\in\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}}\nabla f(\mathcal{X}_c^{(t)},\hat{\boldsymbol{\theta}}_k^{(t)}). \tag{43}$$

Following the same logic outlined in Lemma 3 in (Ghosh et al., 2020), for any $k'\in[K]$ such that $k'\neq k$, we decompose $\sum_{c\in\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}}\nabla f(\mathcal{X}_c^{(t)},\hat{\boldsymbol{\theta}}_k^{(t)})$ into two distinct terms as follows.

$$\sum_{c\in\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}}\nabla f(\mathcal{X}_c^{(t)},\hat{\boldsymbol{\theta}}_k^{(t)})=|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|\nabla F(\mathcal{D}_{k_{c'}^*},\hat{\boldsymbol{\theta}}_k^{(t)})+\sum_{c\in\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}}\left(\nabla f(\mathcal{X}_c^{(t)},\hat{\boldsymbol{\theta}}_k^{(t)})-\nabla F(\mathcal{D}_{k_{c'}^*},\hat{\boldsymbol{\theta}}_k^{(t)})\right). \tag{44}$$

We can establish an upper bound for the l_2 norm of (44) as follows.

$$\left\|\,|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|\nabla F(\mathcal{D}_{k_{c'}^*},\hat{\boldsymbol{\theta}}_k^{(t)})+\sum_{c\in\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}}\left(\nabla f(\mathcal{X}_c^{(t)},\hat{\boldsymbol{\theta}}_k^{(t)})-\nabla F(\mathcal{D}_{k_{c'}^*},\hat{\boldsymbol{\theta}}_k^{(t)})\right)\right\| \tag{45}$$

$$\leq|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|\left\|\nabla F(\mathcal{D}_{k_{c'}^*},\hat{\boldsymbol{\theta}}_k^{(t)})\right\|+\left\|\sum_{c\in\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}}\left(\nabla f(\mathcal{X}_c^{(t)},\hat{\boldsymbol{\theta}}_k^{(t)})-\nabla F(\mathcal{D}_{k_{c'}^*},\hat{\boldsymbol{\theta}}_k^{(t)})\right)\right\| \tag{46}$$

$$\leq|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|L\left\|\hat{\boldsymbol{\theta}}_k^{(t)}-\boldsymbol{\theta}_{c'}^*\right\|+\left\|\sum_{c\in\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}}\left(\nabla f(\mathcal{X}_c^{(t)},\hat{\boldsymbol{\theta}}_k^{(t)})-\nabla F(\mathcal{D}_{k_{c'}^*},\hat{\boldsymbol{\theta}}_k^{(t)})\right)\right\| \tag{47}$$

where the inequality between (45) and (46) is satisfied by the triangle inequality of the norm and the inequality between (46) and (47) is satisfied by the L-smoothness of the population loss function.

Furthermore, leveraging the bounded variance assumption 2 and the independence of minibatch sampling, we can establish that

$$\mathbb{E}\left\|\sum_{c\in\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}}\left(\nabla f(\mathcal{X}_c^{(t)},\hat{\boldsymbol{\theta}}_k^{(t)})-\nabla F(\mathcal{D}_{k_{c'}^*},\hat{\boldsymbol{\theta}}_k^{(t)})\right)\right\|^2\leq|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|\frac{v^2}{b}. \tag{48}$$

Let us consider any $\delta_2\in(0,1)$. Utilizing the Markov inequality, we can establish an upper bound for (47) with probability at least $1-\delta_2$, as follows.

$$|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|L\left\|\hat{\boldsymbol{\theta}}_k^{(t)}-\boldsymbol{\theta}_{c'}^*\right\|+\left\|\sum_{c\in\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}}\left(\nabla f(\mathcal{X}_c^{(t)},\hat{\boldsymbol{\theta}}_k^{(t)})-\nabla F(\mathcal{D}_{k_{c'}^*},\hat{\boldsymbol{\theta}}_k^{(t)})\right)\right\| \tag{49}$$

$$\leq|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|L\|\hat{\boldsymbol{\theta}}_k^{(t)}-\boldsymbol{\theta}_{c'}^*\|+\frac{\sqrt{|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|}v}{\delta_2\sqrt{b}}\leq|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|2L\omega+\frac{\sqrt{|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|}v}{\delta_2\sqrt{b}}. \tag{50}$$

Now we describe the upper bound of the l_2 norm of (43) by using (50). With probability at least $1-(K-1)\delta_2$, we have

$$\left\|\frac{\alpha}{C}\sum_{c\in\bar{\mathcal{S}}_k^*\cap\hat{\mathcal{S}}_k^{(t)}}\nabla f(\mathcal{X}_c^{(t)},\hat{\boldsymbol{\theta}}_k^{(t)})\right\| \leq \frac{\alpha}{C}\sum_{k'\neq k}\left\|\sum_{c\in\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}}\nabla f(\mathcal{X}_c^{(t)},\hat{\boldsymbol{\theta}}_k^{(t)})\right\| \tag{51}$$

$$\leq \frac{\alpha}{C}\sum_{k'\neq k}\left[|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|2L\omega+\frac{\sqrt{|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|}v}{\delta_2\sqrt{b}}\right] \tag{52}$$

$$= \frac{2L\omega\alpha}{C}\sum_{k'\neq k}|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|+\frac{v\alpha}{C\delta_2\sqrt{b}}\sum_{k'\neq k}\sqrt{|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|}. \tag{53}$$

By using (18), for any k', we have $\sum_{k'\neq k}|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|\leq|\mathcal{W}^{(t)}|\leq\frac{128K\|\mathbf{E}^{(t-1)}\|^2}{\Delta_g^2}$. In addition, $\sum_{k'\neq k}\sqrt{|\mathcal{S}_{k'}^*\cap\hat{\mathcal{S}}_k^{(t)}|}\leq\sqrt{(K-1)\frac{128K\|\mathbf{E}^{(t-1)}\|^2}{\Delta_g^2}}$ from the Cauchy-Schwarz inequality. With this, we can establish an upper bound for (53) as follows.

$$\left\|\frac{\alpha}{C}\sum_{c\in\bar{\mathcal{S}}_k^*\cap\hat{\mathcal{S}}_k^{(t)}}\nabla f(\mathcal{X}_c^{(t)},\hat{\boldsymbol{\theta}}_k^{(t)})\right\| \leq \frac{256L\omega\alpha K\|\mathbf{E}^{(t-1)}\|^2}{C\Delta_g^2}+\frac{v\alpha}{C\delta_2\sqrt{b}}\sqrt{(K-1)\frac{128K\|\mathbf{E}^{(t-1)}\|^2}{\Delta_g^2}}. \tag{54}$$

Recall that we have assumed that $\mathcal{E}_1$ holds with probability at least $1-\frac{\delta}{3}$. Consider any $\delta_1,\delta_2,\frac{\delta}{3}\in(0,1)$ such that $\delta_1+(K-1)\delta_2+\frac{\delta}{3}\in(0,1)$. Combining (36), (41), and (54), it is a fact that the following inequality holds with probability at least $1-\delta_1-(K-1)\delta_2-\frac{\delta}{3}$.

$$\|\hat{\boldsymbol{\theta}}_k^{(t+1)}-\boldsymbol{\theta}_k^*\| \leq \sqrt{1-\frac{3\rho\mu L}{(\mu+L)^2}}\|\hat{\boldsymbol{\theta}}_k^{(t)}-\boldsymbol{\theta}_k^*\|+\frac{2v}{\delta_1(\mu+L)\sqrt{b(C\rho-\frac{128K\|\mathbf{E}^{(t-1)}\|^2}{\Delta_g^2})}} \tag{55}$$

$$+\frac{256L\omega\alpha K\|\mathbf{E}^{(t-1)}\|^2}{C\Delta_g^2}+\frac{v\alpha}{C\delta_2\sqrt{b}}\sqrt{(K-1)\frac{128K\|\mathbf{E}^{(t-1)}\|^2}{\Delta_g^2}}. \tag{56}$$

By substituting $\alpha=\frac{2}{\mu+L}$ and $K-1\leq K$, we have

$$\|\hat{\boldsymbol{\theta}}_k^{(t+1)}-\boldsymbol{\theta}_k^*\| \leq \sqrt{1-\frac{3\rho\mu L}{(\mu+L)^2}}\|\hat{\boldsymbol{\theta}}_k^{(t)}-\boldsymbol{\theta}_k^*\|+\frac{2v}{\delta_1(\mu+L)\sqrt{b(C\rho-\frac{128K\|\mathbf{E}^{(t-1)}\|^2}{\Delta_g^2})}}$$
$$+\frac{512L\omega K\|\mathbf{E}^{(t-1)}\|^2}{(\mu+L)C\Delta_g^2}+\frac{Kv\|\mathbf{E}^{(t-1)}\|2\sqrt{128}}{(\mu+L)\Delta_g C\delta_2\sqrt{b}}. \tag{57}$$

By (76) in Lemma 3, we have an upper bound for the spectral norm of the SGN matrix as $\|\mathbf{E}^{(t-1)}\|\leq\frac{6\zeta' C}{\delta}\left(\frac{v^2}{T_a b}\right)^{\frac{1}{2}}$.

Substituting it to (57), we have

$$\|\hat{\boldsymbol{\theta}}_k^{(t+1)} - \boldsymbol{\theta}_k^*\| \leq \sqrt{1 - \frac{3\rho\mu L}{(\mu+L)^2}}\|\hat{\boldsymbol{\theta}}_k^{(t)} - \boldsymbol{\theta}_k^*\| + \frac{2v}{\delta_1(\mu+L)\sqrt{bC\rho(1 - \frac{(48\sqrt{2}\zeta')^2 KCv^2}{\rho\Delta_g^2\delta^2 T_a b})}}$$
$$+ \frac{(96\sqrt{2}\zeta')^2 LwKCv^2}{(\mu+L)\Delta_g^2\delta^2 T_a b} + \frac{96\sqrt{2}\zeta' Kv^2}{(\mu+L)\delta_2\delta\Delta_g\sqrt{T_a}b} \tag{58}$$
$$\leq \sqrt{1 - \frac{3\rho\mu L}{(\mu+L)^2}}\|\hat{\boldsymbol{\theta}}_k^{(t)} - \boldsymbol{\theta}_k^*\| + \frac{2v}{\delta_1(\mu+L)\sqrt{bC\rho(1 - \frac{\lambda KCv^2}{4\rho\Delta_g^2\delta^2 T_a b})}}$$
$$+ \frac{\lambda LwKCv^2}{(\mu+L)\Delta_g^2\delta^2 T_a b} + \frac{\sqrt{\lambda}Kv^2}{(\mu+L)\delta_2\delta\Delta_g\sqrt{T_a}b}. \tag{59}$$

To further simplify the notations, we set $\delta = \delta_1 + (K-1)\delta_2 + \delta_3 \in (0,1)$ such that $\delta_1 = \frac{\delta}{3}$, $\delta_2 = \frac{\delta}{3(K-1)}$, and $\delta_3 = \frac{\delta}{3}$. Finally, we have

$$\|\hat{\boldsymbol{\theta}}_k^{(t+1)} - \boldsymbol{\theta}_k^*\| \leq \sqrt{1 - \frac{3\rho\mu L}{(\mu+L)^2}}\|\hat{\boldsymbol{\theta}}_k^{(t)} - \boldsymbol{\theta}_k^*\| + \epsilon^{(t)} \tag{60}$$

where per-iteration contraction error rate $\epsilon^{(t)}$ as specified in (60) can be expressed as follows.

$$\epsilon^{(t)} = \frac{6v}{\delta(\mu+L)\sqrt{bC\rho(1 - \frac{\lambda KCv^2}{4\rho\Delta_g^2\delta^2 T_a b})}} + \frac{\lambda LwKCv^2}{(\mu+L)\Delta_g^2\delta^2 T_a b} + \frac{3\sqrt{\lambda}K^2v^2}{(\mu+L)\delta^2\Delta_g\sqrt{T_a}b}. \tag{61}$$

Through Lemma 3, we have shown that as t increases, CFL-GP can more accurately cluster a larger number of clients. Consistently, (61) demonstrates that the error rate in updating the client-plurality model towards the desired optimal model decreases with increasing t. This indicates a reduction in the noise introduced by incorrect clustering during model updates as t grows. Notably, in (Ghosh et al., 2020), it has been established that the optimal achievable error rate order in the CFL framework is $\tilde{\mathcal{O}}(1/\sqrt{bC})$ when the central unit (CU) has complete knowledge of the true distribution identities of all clients. This aligns with our result from Equation (61), as $t \to \infty$ ($T_a \to \infty$), all clients will eventually be correctly clustered, and the order of ϵ becomes $\tilde{\mathcal{O}}(1/\sqrt{bC})$.

B.3. Proof of Lemma 3

When $t \geq T_a KP$, where T_a is a natural number, each k-th block row of the gradient profile matrix $\mathbf{G}$ has undergone a minimum of T_a updates. Specifically, this condition ensures that the set $\mathcal{T}_k(t)$, which represents the set of times at which the k-th block row is updated, satisfies $|\mathcal{T}_k(t)| \geq T_a$. To maintain clarity in our exposition, let us recall the decomposition of $\mathbf{G}$, $\mathbf{G} = \mathbf{A}^{(t-1)} + \mathbf{E}^{(t-1)}$, as follows.

$$\mathbf{A}^{(t-1)} = \begin{pmatrix} \sum_{t_1 \in \mathcal{T}_1(t)} \frac{\nabla F(\mathcal{D}_{k_1^*}, \boldsymbol{\theta}_1^{(t_1)})}{|\mathcal{T}_1(t)|} & \cdots & \sum_{t_1 \in \mathcal{T}_1(t)} \frac{\nabla F(\mathcal{D}_{k_C^*}, \boldsymbol{\theta}_1^{(t_1)})}{|\mathcal{T}_1(t)|} \\ \vdots & \ddots & \vdots \\ \sum_{t_K \in \mathcal{T}_K(t)} \frac{\nabla F(\mathcal{D}_{k_1^*}, \boldsymbol{\theta}_C^{(t_K)})}{|\mathcal{T}_K(t)|} & \cdots & \sum_{t_K \in \mathcal{T}_K(t)} \frac{\nabla F(\mathcal{D}_{k_C^*}, \boldsymbol{\theta}_C^{(t_K)})}{|\mathcal{T}_K(t)|} \end{pmatrix} \tag{62}$$

$$\mathbf{E}^{(t-1)} = \begin{pmatrix} \sum_{t_1 \in \mathcal{T}_1(t)} \frac{\mathbf{e}_1(\boldsymbol{\theta}_1^{(t_1)})}{|\mathcal{T}_1(t)|} & \cdots & \sum_{t_1 \in \mathcal{T}_1(t)} \frac{\mathbf{e}_C(\boldsymbol{\theta}_1^{(t_1)})}{|\mathcal{T}_1(t)|} \\ \vdots & \ddots & \vdots \\ \sum_{t_K \in \mathcal{T}_K(t)} \frac{\mathbf{e}_1(\boldsymbol{\theta}_K^{(t_K)})}{|\mathcal{T}_K(t)|} & \cdots & \sum_{t_K \in \mathcal{T}_K(t)} \frac{\mathbf{e}_C(\boldsymbol{\theta}_K^{(t_K)})}{|\mathcal{T}_K(t)|} \end{pmatrix}. \tag{63}$$

Consider the c-th column of the matrix $\mathbf{E}^{(t-1)}$ as $\mathbf{E}_{:,c}^{(t-1)}$, which is a Kd-sized vector. We denote the covariance matrix of the random vector $\mathbf{E}_{:,c}^{(t-1)}$ as $\boldsymbol{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}}$, which can be represented as follows.

$$\boldsymbol{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}} = \begin{pmatrix} \boldsymbol{\Sigma}_{(c,1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{(c,2)} & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\Sigma}_{(c,K)} \end{pmatrix}, \tag{64}$$

where k-th diagonal block $\boldsymbol{\Sigma}_{(c,k)}$ indicates the covariance matrix of the random vector $\sum_{t_k \in \mathcal{T}_k(t)} \frac{\mathbf{e}_c(\boldsymbol{\theta}_k^{(t_k)})}{|\mathcal{T}_k(t)|}$. Note that the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}}$ is a block diagonal matrix, as the K vectors comprising the c-th column of the noise matrix are independent of each other.

To obtain an upper bound on $\mathbb{E}[\|\mathbf{E}^{(t-1)}\|]$, we employ a recent result by Bandeira et al. (Bandeira et al., 2021) on non-asymptotic matrix concentration inequalities.

Lemma 5 (Application of Lemma 3.8 in (Bandeira et al., 2021))**.** *Assume that the columns of a given random matrix $\mathbf{E}^{(t-1)}$ are independent centered Gaussian random vectors. Then, for any $\epsilon > 0$, the expected spectral norm of the matrix $\mathbf{E}^{(t-1)}$ is upper bounded as follows.*

$$\mathbb{E}[\|\mathbf{E}^{(t-1)}\|] \leq (1+\epsilon)\left(\left\| \sum_{c=1}^{C} \boldsymbol{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}} \right\|^{\frac{1}{2}} + \max_c [\mathrm{Tr}(\boldsymbol{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}})]^{\frac{1}{2}} \right) + \frac{\zeta}{\epsilon} \max_c \|\boldsymbol{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}}\|^{\frac{1}{2}} (\log Kd)^{\frac{3}{2}}, \tag{65}$$

where $\boldsymbol{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}}$ is the covariance matrix of the c-th column of $\mathbf{E}^{(t-1)}$ and ζ is a universal constant.

To provide an upper bound of (65), we first explore an upper bound of the spectral norm of $\boldsymbol{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}}$. We have the following inequalities.

$$\left\| \boldsymbol{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}} \right\| = \max_k \left\| \boldsymbol{\Sigma}_{(c,k)} \right\| \leq \max_k \left\| \boldsymbol{\Sigma}_{(c,k)} \right\|_\infty. \tag{66}$$

The equality of (66) holds since $\boldsymbol{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}}$ is a block diagonal matrix and the inequality holds due to the symmetricity of the covariance matrix as $\left\| \boldsymbol{\Sigma}_{(c,k)} \right\|^2 \leq \left\| \boldsymbol{\Sigma}_{(c,k)} \right\|_1 \left\| \boldsymbol{\Sigma}_{(c,k)} \right\|_\infty$ and $\left\| \boldsymbol{\Sigma}_{(c,k)} \right\|_1 = \left\| \boldsymbol{\Sigma}_{(c,k)} \right\|_\infty$.

Furthermore, $\left\| \boldsymbol{\Sigma}_{(c,k)} \right\|_\infty$ is upper bounded as follows.

$$\left\| \boldsymbol{\Sigma}_{(c,k)} \right\|_\infty \leq \frac{1}{|\mathcal{T}_k(t)|^2} \left\| \sum_{t_k \in \mathcal{T}_k(t)} \boldsymbol{\Sigma}_c(\boldsymbol{\theta}_k^{(t_k)}) \right\|_\infty \leq \frac{\max_{t_k} \left\| \boldsymbol{\Sigma}_c(\boldsymbol{\theta}_k^{(t_k)}) \right\|_\infty}{|\mathcal{T}_k(t)|} \leq \frac{\max_{t_k} \mathrm{Tr}\big(\boldsymbol{\Sigma}_c(\boldsymbol{\theta}_k^{(t_k)})\big)}{|\mathcal{T}_k(t)|}, \tag{67}$$

where the first inequality holds because the covariance matrix of the sum of independent random vectors can be expressed as the sum of their individual covariance matrices. The second inequality holds due to the sub-additive property of the matrix infinity norm, and the third inequality holds due to the assumption of the SGN variance. Since the trace of the covariance matrix is upper bounded by $\frac{v^2}{b}$, we finally have

$$\left\|\mathbf{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}}\right\| \leq \frac{v^2}{|\mathcal{T}_k(t)|b} \leq \frac{v^2}{T_a b}. \tag{68}$$

Secondly, we analyze the cumulative spectral norm of the covariance matrices. Leveraging the sub-additivity property of the matrix norm, it follows that

$$\left\|\sum_{c=1}^{C} \mathbf{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}}\right\| \leq \sum_{c=1}^{C} \left\|\mathbf{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}}\right\| \leq C\text{max}_c \left\|\mathbf{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}}\right\| \leq \frac{Cv^2}{T_a b} \tag{69}$$

where the last inequality holds by (68).

The trace of the given covariance matrix $\mathbf{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}}$ can be represented as follows.

$$\text{Tr}\left(\mathbf{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}}\right) = \sum_{k=1}^{K} \text{Tr}(\mathbf{\Sigma}_{(c,k)}) = \sum_{k=1}^{K} \text{Tr}\left(\frac{1}{|\mathcal{T}_k(t)|^2} \sum_{t_k \in \mathcal{T}_k(t)} \mathbf{\Sigma}_c(\boldsymbol{\theta}_k^{(t_k)})\right). \tag{70}$$

Moreover, we have

$$\sum_{k=1}^{K}\left(\frac{1}{|\mathcal{T}_k(t)|^2} \sum_{t_k \in \mathcal{T}_k(t)} \text{Tr}\left(\mathbf{\Sigma}_c(\boldsymbol{\theta}_k^{(t_k)})\right)\right) \leq \sum_{k=1}^{K}\left(\frac{v^2}{|\mathcal{T}_k(t)|b}\right) \leq \frac{Kv^2}{T_a b}, \tag{71}$$

where the first inequality is established under the assumption of bounded variance. Based on (70) and (71), we have

$$\text{Tr}\left(\mathbf{\Sigma}_{\mathbf{E}_{:,c}^{(t-1)}}\right) \leq \frac{Kv^2}{T_a b}. \tag{72}$$

Combining (68), (69), and (72) for (65) with choosing $\epsilon = 1$, we have

$$\mathbb{E}\|\mathbf{E}^{(t-1)}\| \leq 2\left(\frac{Cv^2}{T_a b}\right)^{\frac{1}{2}} + 2\left(\frac{Kv^2}{T_a b}\right)^{\frac{1}{2}} + \zeta\left(\frac{v^2}{T_a b}\right)^{\frac{1}{2}} (\log Kd)^{\frac{3}{2}}. \tag{73}$$

Consider any $\delta \in (0,1)$. Then by the Markov inequality, we have $P(\|\mathbf{E}^{(t-1)}\| \leq \mathbb{E}\|\mathbf{E}^{(t-1)}\|/(\delta)) \geq 1-\delta$. It indicates that the following inequalities hold with probability at least $1-\delta$.

$$\|\mathbf{E}^{(t-1)}\| \leq \frac{\mathbb{E}\|\mathbf{E}^{(t-1)}\|}{\delta} \leq \frac{1}{\delta}\left(2\left(\frac{Cv^2}{T_a b}\right)^{\frac{1}{2}} + 2\left(\frac{Kv^2}{T_a b}\right)^{\frac{1}{2}} + \zeta\left(\frac{v^2}{T_a b}\right)^{\frac{1}{2}} (\log Kd)^{\frac{3}{2}}\right) \tag{74}$$

$$= \frac{1}{\delta}\left(\frac{v^2}{T_a b}\right)^{\frac{1}{2}}\left(2\sqrt{C} + 2\sqrt{K} + \zeta(\log Kd)^{\frac{3}{2}}\right) \tag{75}$$

$$\leq \frac{\zeta' C}{\delta}\left(\frac{v^2}{T_a b}\right)^{\frac{1}{2}} \tag{76}$$

where $\zeta' = 4 + \zeta$. The first inequality in (74) holds due to the Markov inequality, and the second inequality holds by the upper bound provided in (73). The inequality between (75) and (76) holds by $(\log Kd)^{\frac{3}{2}} \leq C$ and $K \leq C$.

B.4. Proof of Proposition 1

Let $\delta \in (0,1)$ and $\delta_1 = \frac{2}{3}\delta$. Under the assumptions outlined in Proposition 1, with probability at least $1-\delta_1$, the number of incorrectly clustered clients, $W(T_{\text{cl}})$, is upper bounded as follows.

$$W(T_{\text{cl}}) \leq \frac{128K(\zeta' Cv)^2}{\Delta_g^2(\frac{2}{3}\delta)^2 T_a b} \tag{77}$$

where $T_a = \lfloor \frac{T_{\text{cl}}-1}{KP} \rfloor$ and ζ' is the universal constant given in Section B.3. We have $T_a = \lfloor \frac{T_{\text{cl}}-1}{KP} \rfloor = \lfloor \frac{\lambda K C^2 v^2}{\delta^2 \Delta_g^2 b} + 1 \rfloor$ which implies $T_a \geq \frac{\lambda K C^2 v^2}{\delta^2 \Delta_g^2 b}$. Substituting this into (77), we obtain $W(T_{\text{cl}}) < 1$. This indicates that for any T, where T is greater than the clustering threshold T_{cl}, the algorithm achieves the optimal clustering with probability at least $1-\delta_1$.

For clarity in subsequent discussions, we define an event $\mathcal{E}_3$ as $\mathcal{E}_3 = \{|W(T_{\text{cl}})| < 1\}$ representing the scenario where the number of incorrectly clustered clients is less than one, effectively indicating correct clustering of all clients. According to the conditions outlined in Proposition 1, the likelihood of $\mathcal{E}_3$ occurring is at least $1-\delta_1$.

This leads to the following interpretation of our analysis: CFL-GP ensures a consistent decrease in the upper bound of incorrectly clustered clients. Therefore, after a substantial number of learning iterations t, CFL-GP achieves the optimal clustering with high probability, especially for $t \geq T_{\text{cl}}$. This scenario permits the direct use of established high-probability convergence results from stochastic gradient descent methodologies.

Building upon this, we adopt the high-probability bound for stochastic gradient descent from (Rakhlin et al., 2012). For the given assumptions for Proposition 1, we can readily observe that $\|\frac{1}{|\mathcal{S}_k^*|}\sum_{c\in\mathcal{S}_k^*} \nabla f(\mathcal{X}_c^{(t)}, \hat{\boldsymbol{\theta}}_k^{(t)})\|^2 \leq (H^2/(C\rho))$ almost surely. Then, we state the following proposition.

Proposition 7 (Application of Proposition 1 in (Rakhlin et al., 2012))**.** *Let $\delta_2 \in (0, 1/e)$. Under the assumptions given for Proposition 1 and suppose the event $\mathcal{E}_3$ holds. Then the following inequality holds with probability at least $1-\delta_2$.*

$$\|\hat{\boldsymbol{\theta}}_k^{(\hat{t}+T_{cl})} - \boldsymbol{\theta}_k^*\| \leq \frac{(624\log(\log(\hat{t})/\delta_2)+1)(H^2/(C\rho))}{\mu^2 \hat{t}} \tag{78}$$

where $\hat{t} \in \mathbb{N}$ and $\hat{t} + T_{\text{cl}} = T$. We define an event $\mathcal{E}_4$ as $\mathcal{E}_4 = \{\|\hat{\boldsymbol{\theta}}_k^{(T)} - \boldsymbol{\theta}_k^*\| \leq \frac{(624\log(\log(T-T_{\text{cl}})/\delta_2)+1)(H^2/(C\rho))}{\mu^2(T-T_{\text{cl}})}\}$. The probability that event $\mathcal{E}_3$ occurs is at least $1-\delta_1$, and conditional on $\mathcal{E}_3$, event $\mathcal{E}_4$ holds with probability at least $1-\delta_2$.

Then we can conclude that under the assumptions given for Proposition 1, $P(\mathcal{E}_4) \geq P(\mathcal{E}_4 \cap \mathcal{E}_3) = P(\mathcal{E}_4|\mathcal{E}_3)P(\mathcal{E}_3)$ and $P(\mathcal{E}_4|\mathcal{E}_3)P(\mathcal{E}_3) \geq 1-\delta_1-\delta_2$. By choosing $\delta_2 = \delta/3 < 1/e$, we can conclude that, with probability at least $1-\delta$, $\mathcal{E}_4$ holds.

C. Extensions for Improving Communication Efficiency

In the main text, we explored a learning protocol where local clients update a model in a single step and then transmit the corresponding gradient to a central unit, often referred to as the gradient-averaging protocol. We can extend CFL-GP to incorporate the model averaging approach, as detailed in Algorithm 4, to enhance communication efficiency. This protocol, originally proposed by (McMahan et al., 2017), has been widely adopted in a variety of federated learning (FL) algorithms (McMahan et al., 2017; Ghosh et al., 2020; Sattler et al., 2020). Within this framework, each client executes multiple local updates on the model received from the central unit (CU) using their own local dataset before transmitting the updated model back to the CU.

Algorithm 4 diverges from Algorithm 1 by adopting a model-averaging strategy for local updates at time step t. In this approach, the k-th model is transmitted to the clients within the specified k-th cluster, whereupon these clients undertake local updates by progressing through mini-batches from their own datasets over $T_L(c)$ iterations. This process follows the LOCALUPDATE protocol as specified in Algorithm 5, with the learning rate for the c-th client denoted by γ_c. After completing the updates, the discrepancy between the locally enhanced model and the initial model from the CU, the model gap $\Delta\boldsymbol{\theta}$, is transmitted back to the CU.

When updating the gradient profile matrix at a time step t satisfying $t, \texttt{mod}, P = 1$ and $t < T_{\text{cl}}$, the $\bar{k}^{(t)}$-th model is broadcasted, and model gaps related to this model are aggregated from the clients. These model gaps are then utilized to

Algorithm 4 CFL-GP with Model Averaging protocol

Input: K initial models $\{\boldsymbol{\theta}_k^{(0)}\}_{k=1}^K$, K initial clusters $\{\mathcal{S}_k^{(0)}\}_{k=1}^K$, clustering period P, learning rate γ, gradient features initialized as $\boldsymbol{g}_c^{(-1)} = \mathbf{0}\ \forall c \in [C]$, feature moving average factor $\{\beta_t\}_{t=1}^T$, initial broadcast model index $\bar{k}^{(0)} = 0$.

Output: K trained models $\{\boldsymbol{\theta}_k^{(T)}\}_{k=1}^K$ and K updated clusters $\{\mathcal{S}_k^{(T)}\}_{k=1}^K$

for $t = 0$ **to** T **do**
  for $k = 1$ **to** K in parallel **do** ▷ *Model Update*
    CU transmits $\boldsymbol{\theta}_k^{(t)}$ to the clients in $\mathcal{S}_k^{(t)}$ and
    Clients feedback $\Delta\boldsymbol{\theta}_{k,c}^{(t)} = \textsc{LocalUpdate}(\boldsymbol{\theta}_k^{(t)}, c)\ \forall c \in \mathcal{S}_k^{(t)}$
    CU updates $\boldsymbol{\theta}_k^{(t+1)} := \boldsymbol{\theta}_k^{(t)} + \frac{1}{|\mathcal{S}_k^{(t)}|}\sum_{c\in\mathcal{S}_k^{(t)}} \Delta\boldsymbol{\theta}_{k,c}^{(t)}$
  if $t \bmod P = 1$ and $t < T_{\text{cl}}$ **then** ▷ *Cluster Update*
    Update $\bar{k}^{(t)} := (\bar{k}^{(t-1)} \bmod K) + 1$ ▷ round-robin manner
    CU broadcasts $\boldsymbol{\theta}_{\bar{k}^{(t)}}^{(t)}$ and receives $\Delta\boldsymbol{\theta}_{\bar{k}^{(t)},c}^{(t)}\ \forall c \in [C]$
    CU updates $\boldsymbol{g}_{c,\bar{k}^{(t)}}^{(t)} := (1-\beta_t)\boldsymbol{g}_{c,\bar{k}^{(t)}}^{(t-1)} + \beta_t \Delta\boldsymbol{\theta}_{\bar{k}^{(t)},c}^{(t)}\ \forall c \in [C]$
    $\boldsymbol{g}_{c,k'}^{(t)} := \boldsymbol{g}_{c,k'}^{(t-1)}\ \forall c \in [C], k' \in [K] s.t. k \neq \bar{k}^{(t)}$
    $\{\mathcal{S}_k^{(t+1)}\}_{k=1}^K := \textsc{SpectralClustering}(\{\boldsymbol{g}_c^{(t)}\}_{c=1}^C, \{\mathcal{S}_k^{(t)}\}_{k=1}^K)$
  else
    $\mathcal{S}_k^{(t+1)} := \mathcal{S}_k^{(t)}\ \forall k \in [K]$ and $\boldsymbol{g}_c^{(t)} := \boldsymbol{g}_c^{(t-1)}\ \forall c \in [C]$

Algorithm 5 LocalUpdate

Input: Model $\boldsymbol{\theta}$, client c

Output: Model gap $\Delta\boldsymbol{\theta}$

$\boldsymbol{\theta}' = \boldsymbol{\theta}$
for $\mathcal{X}_c$ in $\{\mathcal{X}_c^{(1)}, ..., \mathcal{X}_c^{(T_L(c))}\}$ **do** ▷ $T_L(c)$ times local update
  $\boldsymbol{\theta}' := \boldsymbol{\theta}' - \gamma_c \nabla f(\mathcal{X}_c, \boldsymbol{\theta}')$
$\Delta\boldsymbol{\theta} = \boldsymbol{\theta}' - \boldsymbol{\theta}$

update the client features instead of the single-step gradients. This methodology enables the implementation of the CFL-GP using a model averaging protocol without necessitating additional changes to its fundamental logic.

D. Computation and Communication Cost

To examine the computation and communication costs associated with CFL algorithms, it is crucial to acknowledge that these algorithms introduce additional clustering operations within the Federated Learning (FL) framework. These operations require gradient information, model performance evaluation, or computations beyond what is traditionally necessary for the FL framework. Thus, the overall costs of CFL algorithms are often significantly affected by the *speed of successful clustering* as once clustering is deemed successful, those resources can be saved without the need for any further clustering steps. For instance, if the client clusters remain unchanged for several iterations, we can conclude that the clustering solution has converged, and focus solely on model updates. Therefore, evaluating the convergence speed of client clustering is crucial, in addition to assessing clustering accuracy.

One of CFL-GP's primary advantages is its capability to ensure fast convergence towards accurate clustering. In the following subsection, we explore the convergence speed of client clustering of CFL algorithms in detail. We will demonstrate that CFL-GP exhibits exceptional performance in achieving optimal clustering and accelerating the speed of clustering convergence.

D.1. Convergence speed towards optimal clustering

Through this subsection, the clustering convergence speed of the CFL algorithms is examined based on the series of experiments described in Appendix F. CFL-GP demonstrates outstanding performance in multiple aspects, notably in clustering speed and accuracy. We have incorporated the following metrics to comprehensively evaluate the clustering

convergence performance of the algorithms.

Metrics.

- The number of communication rounds a CFL algorithm requires to reach the peak ARI (Adjusted Rand Index) observed during the training phase.
- The number of communication rounds needed to attain the ideal ARI score of 1.0.

For instances where a CFL algorithm achieves a specific ARI value but does not realize optimal clustering, we label this outcome as "-," indicating the inability to attain optimal clustering. We measure performance using 5 to 20 distinct random seeds per experiment. In scenarios where a CFL algorithm repeatedly fails to achieve optimal clustering across multiple trials but manages it at least once, we record numerical values rather than "-." Here, we estimate the communication rounds necessary for optimal clustering by considering the highest count of rounds where the CFL algorithm falls short of finding the optimal clustering. Moreover, situations where a CFL algorithm fails to detect any clusters are marked as "Clustering Fail."

Instances of a CFL algorithm achieving the optimal ARI are emphasized in **bold**. Additionally, in cases where several CFL algorithms reach the optimal ARI, we underscore the one that does so in the least number of communication rounds.

Table 3. Convergence speed of clustering towards maximum ARI and optimal clustering. CFL-GP demonstrates convergence that is **up to more than 100× faster with a 100% success rate**.

	[Exp in F.5] t of Achieving Maximum ARI / t of Achieving Optimal ARI				
Algorithm	C=8	C=16	C=32	C=64	C=128
CFL-GP	**3.67±5.73 / 3.67±5.73**	**1.00±0.00 / 1.00±0.00**	**1.00±0.00 / 1.00±0.00**	**1.00±0.00 / 1.00±0.00**	**1.00±0.00 / 1.00±0.00**
IFCA	7.00±3.62 / -	12.56±6.06 / 137.22±90.20	12.00±4.71 / 73.67±90.06	15.33±7.57 / 76.56±88.21	13.22±4.80 / 55.56±77.88
MADMO	82.11±75.43 / -	134.67±28.87 / -	135.56±31.67 / -	125.89±32.32 / -	99.78±7.94 / -

	[Exp in F.5] t of Achieving Maximum ARI / t of Achieving Optimal ARI			
Algorithm	b=32	b=64	b=128	b=256
CFL-GP	**65.67±11.70 / 65.67±11.70**	**1.00±0.00 / 1.00±0.00**	**1.00±0.00 / 1.00±0.00**	**1.00±0.00 / 1.00±0.00**
IFCA	24.22±8.34 / -	12.00±4.71 / 73.67±90.06	10.44±3.20 / 74.22±89.69	9.78±5.61 / 115.44±95.71
MADMO	73.33±28.89 / -	135.56±31.67 / -	Clustering Fail	Clustering Fail

	[Exp in F.7] t of Achieving Maximum ARI / t of Achieving Optimal ARI		
Algorithm	C=20	C=40	C=80
CFL-GP	3.00±1.41 / 3.00±1.41	**3.40±1.74 / 3.40±1.74**	5.30±2.78 / 5.30±2.78
IFCA	**2.95±0.50 / 2.95±0.50**	3.75±0.62 / 3.75±0.62	**4.60±0.49 / 4.60±0.49**
MADMO	126.95±96.00 / -	106.55±102.12 / -	Clustering Fail

	[Exp in F.8] t of Achieving Maximum ARI / t of Achieving Optimal ARI		
Algorithm	C=10	C=80	C=160
CFL-GP	**1.00±0.00 / 1.00±0.00**	**2.71±0.45 / 2.71±0.45**	**7.43±1.84 / 7.43±1.84**
IFCA	2.43±0.90 / 31.00±44.27	5.86±2.23 / 20.14±33.03	11.43±1.76 / 11.43±1.76
MADMO	22.00±21.51 / 64.86±33.34	8.14±1.64 / 20.86±32.72	14.00±6.23 / -

Table 4. Convergence speed of clustering towards maximum ARI and optimal clustering on the wireless Channel compression problems, CD1 and CD2. CFL-GP achieves the optimal clustering much faster.

	[Exp in F.6] t of Achieving Maximum ARI / t of Achieving Optimal ARI				
Algorithm	CD1 C=16	CD1 C=32	CD2 C=10	CD2 C=20	CD2 C=40
CFL-GP	**1.00±0.00 / 1.00±0.00**	**1.00±0.00 / 1.00±0.00**	**120.33±8.38 / 120.33±8.38**	**195.00±46.68 / 195.00±46.68**	**267.00±12.96 / 267.00±12.96**
IFCA	Clustering Fail	Clustering Fail	Clustering Fail	Clustering Fail	Clustering Fail
MADMO	890.50±889.50 / 1890.50±110.50	1651.00±2.00 / -	111.00±17.11 / -	453.33±207.36 / -	429.00±310.32 / -

Clustering performance and convergence speed. As demonstrated in Table 3, CFL-GP exhibits remarkable efficiency in achieving optimal clustering, sometimes requiring only a single round of spectral clustering in specific experiments (see **[Exp in F.5]** with $C = 16$ to 128 and $b = 64$ to 256). This efficiency highlights CFL-GP's capability to accurately identify

the true cluster identity of clients through a single spectral clustering step, in stark contrast to other CFL algorithms. The baseline algorithms often fail to reach optimal clustering even after 100 communication rounds and take significantly longer to achieve their highest ARI without ever attaining the ideal value of 1.0.

CFL-GP's consistency in achieving strong clustering convergence is further evidenced in datasets involving models with over a million parameters, as indicated in Table 4. Its performance in client clustering tasks using the wireless channel model datasets, achieving optimal clustering in just one round, is especially notable. In experiments on CD2 for C=10, 20, and 40 (shown in Table 4), CFL-GP identified optimal clusters first within 270 communication rounds, in contrast to other CFL algorithms which failed to achieve a 1.0 ARI.

CFL-GP's rapid convergence speed thus reduces computation and communication costs in many experiments, thereby substantially lightening the client clustering burden. Additional analysis of ARI values and task performance relative to communication rounds is provided in Section F.

D.2. Communication and computation cost before clustering convergence

We discuss three main cost types: communication costs, gradient/loss computation costs, and spectral clustering costs, especially for CFL-GP.

Communication cost. Table 5 details the communication costs incurred before the completion of clustering. These costs are quantified by d, denoting the number of model parameters that must be exchanged between the CU and a client. *Downlink* describes the transmission of models from the CU to clients, and *uplink* signifies the transmission from clients to the CU. Specifically, within the interval from $t = nP$ to $(n+1)P - 1$, the communication complexity experienced by each client/CU is illustrated in Table 5.

Table 5. Communication cost before clustering convergence

Algorithm	Downlink (CU to client)	Uplink (client to CU)
CFL-GP	$Y(< P+1)$	$Y(< P+1)$
IFCA	KP	P
MADMO	P	P
FedAvg	P	P

We denote the communication complexity of CFL-GP as Y. Note that Y cannot be defined as a closed form since CFL-GP's complexity varies over iteration and the clients' estimated cluster identities. However, we can obtain an upper bound of the CFL-GP's communication complexity as $P + 1$. At each interval of P, the CU selects one model and broadcasts it to the clients. As a result, the downlink communication complexity becomes, at most, $P + 1$. However, if there are clients with estimated cluster identities corresponding to the index of the broadcast model, we do not need to re-transmit the broadcast model to them. Therefore, the overall average downlink communication cost for a given client is less than $P + 1$. Although Y may vary for each iteration, its upper bound is $P + 1$.

Gradient and loss computation cost. Table 6 displays the frequency of gradient/loss computations performed by each CFL algorithm over P communication rounds. For CFL-GP, this gradient computation cost is represented by Y. In each P cycle, clients are sent one or two models from the central unit, on which they compute gradients using their local data. Consequently, they may conduct up to $P + 1$ gradient computations in each P interval. However, if a client's estimated cluster identity matches the index of the model sent by the central unit, they only need to perform P gradient computations, as no additional models are received. This sets the maximum gradient computation cost for CFL-GP at $P + 1$. Clients over IFCA are required to receive all models from the central unit to perform clustering and evaluate the performance achievable with these models, involving model forward propagation. CFL-GP, MADMO, and the Global model approach (FedAvg) eliminate the need for such additional procedures.

Table 6. Gradient and loss computation cost before clustering convergence

Algorithm	Gradient computation	Loss computation
CFL-GP	$Y(< P+1)$	0
IFCA	P	C
MADMO	P	0
FedAvg	P	0

CFL-GP distinguishes itself by achieving clustering convergence significantly faster than alternative approaches. This rapid convergence allows for a substantial reduction in computation and communication costs, as the clustering updates can be terminated once clustering convergence is observed.

Computation cost of spectral clustering. The spectral clustering process of CFL-GP is performed on the gradient profile matrix $\mathbf{G}^{(t)} = [\boldsymbol{g}_1^{(t)}, \cdots, \boldsymbol{g}_C^{(t)}] \in \mathbb{R}^{Kd \times C}$. This process consists of two main steps. The first step is to recover K leading singular vectors $\hat{\boldsymbol{U}} \in \mathbb{R}^{Kd \times K}$, and the second step is to perform K-means clustering on the reduced feature matrix using those leading singular vectors. Recent advances in principal component analysis and leading singular vector recovery techniques, such as those presented in (Shamir, 2015), have enabled rapid solutions to this process. Additional iterations are performed to achieve the required accuracy, with the runtime being logarithmic in the required accuracy (Shamir, 2016). Furthermore, when dealing with large d (the number of parameters in a model), gradient compression can be successfully employed, as discussed in Section E, due to the low intrinsic dimensionality of neural networks. K-means clustering problem is an NP-hard problem. However, for a given fixed number of iterations t_k for K-means clustering, the complexity is $\mathcal{O}(t_k C K^2)$ with an initialization technique (Arthur & Vassilvitskii, 2007) which is negligible as K is the total number of clusters.

Discussion. CFL-GP has consistently demonstrated lower costs in comparison to other CFL algorithms across various experiments. This efficiency primarily stems from the fact that the major costs associated with CFL algorithms are due to the pursuit of optimal clustering. CFL-GP excels by reaching this optimal clustering markedly faster—often 100 times quicker—than its counterparts. Consequently, CFL-GP can reduce the resources expended on clustering.

CFL-GP also offers theoretical assurances for attaining optimal clustering with a high probability, setting it apart from other CFL algorithms. Experimental evidence further corroborates its high success rate in achieving optimal clustering, underscoring CFL-GP's effective resource utilization by promptly reaching optimal clustering and permitting an early halt to the clustering updates, thereby dedicating resources primarily to model updates.

Moreover, as detailed in Appendix E, the computational burden of spectral clustering can be substantially alleviated through the use of compressed gradients for constructing the gradient profile matrix. Remarkably, CFL-GP maintains high clustering performance even with compression ratios up to 10^4, outperforming baseline algorithms in some scenarios where the baselines struggle to achieve effective clustering. This adaptability highlights CFL-GP's suitability for large parameterized models, including neural networks, as further elaborated in the scalability section of Appendix E.

E. Compressed or selected gradient information

We demonstrate that CFL-GP achieves optimal clustering performance even with significantly compressed or selectively chosen gradient information in some scenarios. Supporting evidence from recent studies suggests that neural network models and their gradients exhibit low intrinsic dimensionality across numerous tasks. Consequently, techniques such as random sampling and projection applied to models or gradients have been leveraged for diverse objectives, as highlighted by (Li et al., 2018; Aghajanyan et al., 2020; Hu et al., 2021a; Vogels et al., 2019).

E.1. Compressed gradient information

In this subsection, we demonstrate that CFL-GP retains optimal clustering performance across certain scenarios, even when limited to a subset of the gradient information. Notably, CFL-GP's efficacy remains uncompromised with a gradient compression ratio as high as 10^3.

Recall the gradient feature update process in Algorithm 1:

$$\boldsymbol{g}_{c,\bar{k}} := (1-\beta_t)\boldsymbol{g}_{c,\bar{k}} + \beta_t \nabla f(\mathcal{X}_c^{(t)}, \boldsymbol{\theta}_{\bar{k}}) \quad \forall c \in [C]. \tag{79}$$

Suppose that we aim to reduce the size of the gradient from d to $\hat{d}$, where $\hat{d} < d$. We define a set of random indices $\mathcal{I} = \{i_1, \cdots, i_{\hat{d}}\}$, with each element selected randomly from $[d]$ without replacement. We then denote the partial gradient information obtained by selecting only the elements whose indices $i \in \mathcal{I}$ as $\hat{\nabla} f(\mathcal{X}_c^{(t)}, \boldsymbol{\theta}_{\bar{k}})$.

Based on this, we have a smaller gradient profile matrix with a size of $K\hat{d} \times C$, and we can update it through

$$\boldsymbol{g}_{c,\bar{k}} := (1-\beta_t)\boldsymbol{g}_{c,\bar{k}} + \beta_t \hat{\nabla} f(\mathcal{X}_c^{(t)}, \boldsymbol{\theta}_{\bar{k}}) \quad \forall c \in [C] \tag{80}$$

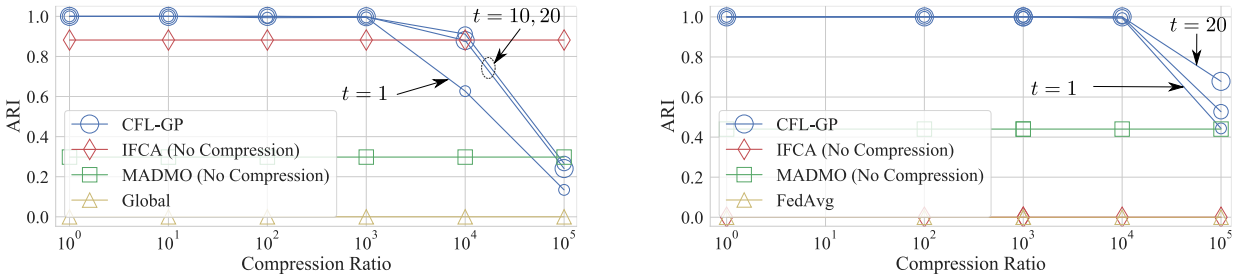

Figure 4. ARI with respect to the gradient compression ratio. (Left): ARI vs. compression ratio on MNIST experiments. Despite utilizing just 1/1,000 of the gradient information (when gradient compression ratio equals to 10^3), the clustering approach of CFL-GP achieves an optimal ARI of 1.0, as shown on the plot with a **logarithmic horizontal axis**. (Right): ARI vs. compression ratio on COST2100 experiments. Even when using only 1/10,000 of the gradient information, CFL-GP outperforms both IFCA and MADMO. This indicates that the gradients are highly informative and reliable for client clustering.

where $\boldsymbol{g}_{c,\bar{k}}$ is now a $\hat{d}$-dimensional vector. We note that we reuse the notation $\boldsymbol{g}$ here for readability, although it is now representing a placeholder for the compressed gradient object than the original full-dimensional gradient profile matrix column. Compression Ratio (CR) can be represented as $\frac{d}{d'}$.

To investigate the impact of compression ratio on the clustering performance of CFL-GP, we use two experimental setups, which are outlined below.

The first experimental setup, detailed in Appendix F.5.1, involves models with over 150,000 parameters. In this setup, our objective is to cluster 80 clients into four distinct groups, with each group assigned to a specific model. The clustering is based on the rotation transformation applied to the MNIST dataset, resulting in four different clusters representing rotations of 0, 90, 180, and 270 degrees.

In the second experimental setup, our focus is on clustering 32 data centers into two clusters based on the heterogeneity of channel distribution in the context of training deep autoencoder models. Further details of the experiment can be found in Section F.6. In this experiment, the model averaging protocol specified in Algorithm 4 is employed for CFL-GP.

We performed experiments with varying compression ratios ranging from 1 to 10^5 for three different gradient profile matrices obtained at $t = 1, 10, 20$. The results for the two experimental setups are depicted in Figure 4. The left subplot presents the results for the first experimental setting, while the right subplot displays the results for the second experimental setting. Each subplot demonstrates the change in clustering performance as the gradient compression ratio is modified. A higher compression ratio indicates a more significant compression of the gradients. The size of the plot marker corresponds to the value of t, with larger markers representing higher values.

Impact of gradient compression ratio on clustering performance. The results depicted in Figure 4 illustrate the impact of varying gradient compression ratios on the clustering performance. The left subplot shows that CFL-GP maintains optimal clustering performance up to a compression ratio of 10^3, indicating that the compressed gradients still contain highly informative features. When the compression ratio exceeds 10^4, CFL-GP starts to lose its clustering optimality. Nevertheless, for higher values of t (specifically, $t = 10$ and $t = 20$) where more gradient information is accumulated, CFL-GP surpasses the maximum achieved Adjusted Rand Index (ARI) by the IFCA, demonstrating its superiority in clustering performance.

In the right side subplot of Figure 4, CFL-GP is observed to maintain an ARI of 0.99 even with a gradient compression ratio of up to 10^4. This result is particularly remarkable as it stands in contrast to the performance of the others, which do not perform optimally to identify proper clusters in this experiment.

These simulation results serve as compelling evidence for the highly informative nature of gradient information. In order to further enhance our intuitive understanding of the effectiveness of gradients in client clustering, we present visualizations of the (reduced) gradient profile matrix constructed by CFL-GP when utilizing compressed gradient information through Figure 5.

Figure 5 illustrates plots obtained from the Rotated MNIST simulation environment, where the four distinct markers represent distribution identities based on rotation transformations (0, 90, 180, and 270 degrees) of the MNIST dataset. Each

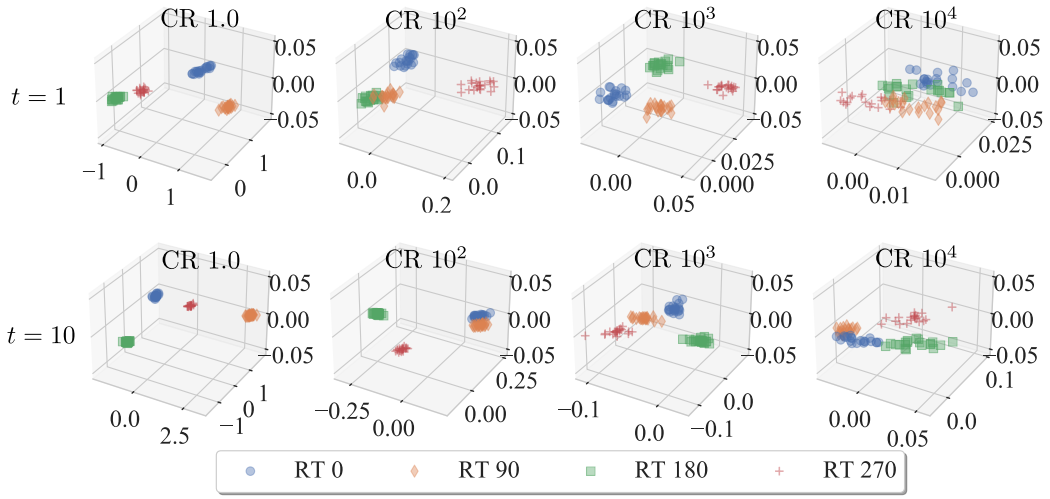

Figure 5. Visualization of the reduced gradient profile matrix, $\hat{\boldsymbol{U}}^\top \mathbf{G}$. Each data point in the subplots represents the reduced gradient feature of each client based on the given gradient Compression Ratios (CR). These features, $\boldsymbol{g}'_c = \hat{\boldsymbol{U}}^\top \boldsymbol{g}_c$, are obtained through the spectral clustering from the compressed gradient profile matrix. The rotation transformations (RT) at angles of 0, 90, 180, and 270 degrees introduce four distinct distributions in the dataset. As the compression ratio increases, client features within the same group become more dispersed. Nevertheless, even at a high compression ratio of 10^4, the clustering performance remains robust. For instance, at $t = 10$, where a significant amount of gradient information is accumulated in the gradient profile matrix, the four clusters are still easily distinguishable, resulting in an ARI of 0.9.

subplot comprises 80 data points, representing the feature vectors of 80 clients. CFL-GP employs spectral clustering with the gradient profile matrix. During this process, the feature vector dimensionality of the c-th client is reduced as $\boldsymbol{g}'_c = \hat{\boldsymbol{U}}^\top \boldsymbol{g}_c$, and each data point corresponds to $\boldsymbol{g}'_c$ (the reduces feature vectors are further randomly projected to 3-dimension space for the visualization purpose). We extract the reduced feature vectors from various gradient profile matrices based on the compression ratio (CR) and t. The first row displays features extracted from the gradient profile matrix at $t = 1$, while the second row corresponds to features extracted at $t = 10$.

Visualization of Gradient profile matrix according to compression ratio. Figure 5 demonstrates that the gradient profile matrix operated by CFL-GP can accurately construct feature vectors capable of distinguishing four clusters within a single clustering period, as shown in the top-left subplot. In contrast, IFCA and MADMO achieve ARI scores of 0.9 and 0.3, respectively. This highlights the reliability and robustness of CFL-GP's clustering criteria.

It is observed that as the gradient compression ratio increases, the scattering of each cluster becomes more pronounced. However, even with a compression ratio of 10^3, the four clusters in the dataset can still be easily distinguished, as depicted in the third row of Figure 5. As the compression ratio is further increased to 10^4, it becomes challenging to achieve optimal clustering solely based on the gradient information obtained within a single clustering period ($t = 1$). Nevertheless, when a greater amount of gradient information is accumulated in the gradient profile matrix ($t = 10$), CFL-GP outperforms IFCA even under the high compression ratio of 10^4. It is clearly observed in the bottom-right plot of the figure, where the gradient profile matrix obtained at $t = 10$, $CR = 10^4$ exhibits more compact clusters compared to the one obtained at $t = 1$, $CR = 10^4$, resulting in a higher ARI score of 0.9.

Decimal point quantization. To further examine the impact of different types of gradient compression, we implement decimal point quantization, where gradient magnitudes predominantly range from 0 to 1. This quantization strategy preserves a specified number of digits (n_d) beyond the decimal point, zeroing out all others. We use the simulation environment using the MNIST dataset previously described, involving models with over 150,000 parameters. The objective is to train multiple models for a classification task on the MNIST dataset. Gradient magnitudes observed during training ranged between -0.22 and 0.13.

Table 7 details the ARI and ACC performance metrics for CFL-GP at varying levels of quantization, alongside the metrics

Table 7. ACC (%) and ARI across varying levels of decimal point quantization

	CFL-GP n_d=0	n_d=1	n_d=2	n_d=3	n_d=4	IFCA	MADMO	Global
ARI	0.00	0.034	1.00	1.00	1.00	0.89	0.31	0.00
ACC	10.0	16.8	87.3	88.7	88.7	85.5	80.0	58.4

for baseline algorithms without quantization

As shown in Table 7, with quantization up to two digits, CFL-GP maintains the highest accuracy compared to baselines without quantization, indeed highlighting our algorithm's robustness to gradient noise. We note that preserving the gradient direction with two digits does not compromise client clustering optimality with 1.0 ARI. Reducing to a single digit after the decimal significantly diminishes gradient information, leading to lower ARI and accuracy. Eliminating all decimal information drops accuracy to 10%, indicating random classification across the ten classes in the environment.

E.2. Selected gradient information

In this subsection, we explore the effectiveness of CFL-GP in clustering clients when it utilizes a gradient profile matrix composed of gradients corresponding to selected parameters, which form a *subset* of the overall model being trained.

There is often a need to customize specific parts of the model for different clusters while maintaining a shared structure across other parts that can benefit all clients. This approach is commonly employed when it is believed that the local datasets owned by the clients have a shared representation. For instance, in the case of clients working with image datasets, they may choose to train a large neural network like ResNet18 and share the deep convolutional layers, which are critical for capturing image features in various applications (Montavon et al., 2018; Yosinski et al., 2014; Goodfellow et al., 2016; LeCun et al., 2015). Simultaneously, they may aim to develop customized classifiers for different client clusters by modifying the last few layers of the model. Such structures, characterized by shared weights and customized outputs, are commonly found in neural network architectures for multi-task learning problems (Ruder, 2017; Collobert & Weston, 2008). The idea of a multi-output architecture in the context of CFL problems was first introduced in (Ghosh et al., 2020).

We consider the scenario in F.7 with 80 clients utilizing ResNet18 models (He et al., 2016) and partial CIFAR-10 data. Notably, half of these clients have permuted labels (for a more detailed setup, see Section F.7). The objective of this task is to cluster the clients while enabling them to share the weights of the feature extraction layers of the ResNet18 model. Simultaneously, the clients should also learn customized last fully connected layers through clustering, which is essential for addressing the challenges posed by permuted labels within each group. CFL-GP constructs the gradient profile matrix using only the gradients corresponding to the last layer of the model.

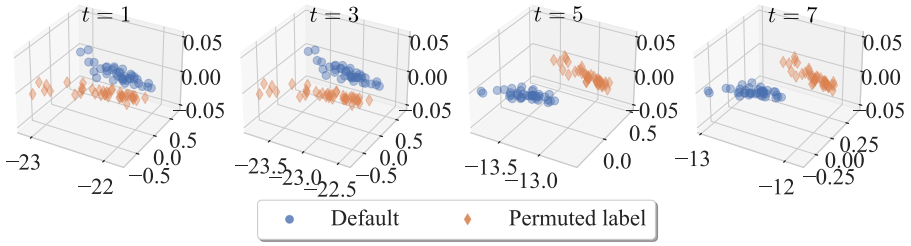

Figure 6. The data points in the subplots correspond to the compressed gradient features of each client, represented as $\boldsymbol{g}'_c = \hat{\boldsymbol{U}}^\top \boldsymbol{g}_c$. These features are derived from the gradient information obtained from the last layer of ResNet18. The clients with local datasets containing label permutations exhibit distinct feature vectors compared to clients with pure subsets of CIFAR-10. This observation emphasizes that even partial gradients can effectively capture the distribution identities of the clients. As the accumulation of gradient information progresses (with increasing t), the gap between the clusters becomes more prominent. In this simulation environment, MADMO yields ARI scores close to 0.0, while CFL-GP achieves a perfect ARI score of 1.0.

Figure 6 visualizes the reduced features of each client at different time steps (t) in the CIFAR-10 experiment. As the gradient profile matrix accumulates more gradient information (i.e., with increasing t), the distance between the two data groups becomes more pronounced. CFL-GP achieves optimal clustering within an average of 5.3 rounds. The results

consistently exhibit an ARI score of 1.0 starting from $t = 5$. In contrast, MADMO could not achieve a positive ARI score, indicating challenges in identifying meaningful clusters under these conditions. This demonstrates the robustness of CFL-GP's gradient-based clustering criteria in detecting clusters, even when only part of the gradient information is utilized. For additional simulation details pertaining to this setup, please refer to Section F.7.

F. Experiment Results

Outline. The detailed experimental setup and configurations are outlined at the beginning of Section F. We delve into an ablation study focusing on the feature update parameter of CFL-GP in Section F.1, showcasing its role in ensuring better Adjusted Rand Index (ARI) outcomes through gradient time-averaging. Section F.2 presents a comparative analysis of CFL-GP's clustering efficacy against the PFL algorithm, as described in (Marfoq et al., 2021), which undertakes implicit clustering. Furthermore, we compare the algorithms' performance with that of a recent CFL algorithm (Vahidian et al., 2023) based on a distinct communication protocol, detailed in Appendix F.3.

We employ a wide range of datasets and experimental settings. The synthetic datasets (F.4) and the MNIST dataset with challenging extensions (F.5) are used to thoroughly analyze the algorithms' sensitivity to factors such as the number of clients, batch size, and distribution similarity.

In Section F.6, we investigate the applicability and scalability of CFL-GP in practical scenarios by conducting experiments using autoencoders and wireless channel datasets characterized by high sample variance. In these challenging environments, CFL-GP achieves optimal clustering and exhibits the highest task performance.

Furthermore, in Section F.7, we demonstrate CFL-GP's ability to achieve optimal clustering even when utilizing selected gradient information in large-scale models like ResNet18 (He et al., 2016). In the additional benchmark experiments using EMNIST (F.8), CFL-GP consistently achieves optimal clustering and high task performance, highlighting its robustness compared to existing CFL algorithms.

The proposed algorithm consistently achieved the highest ARI values among all CFL algorithms in all the experiments, thereby demonstrating superior task performance. In addition to the quantitative results, we provide visualizations of the clients' gradient features obtained through CFL-GP's spectral clustering process in specific experiments. It serves to validate the highly informative nature of gradients and provide additional evidence supporting the robustness of CFL-GP.

Configurations and Fairness. For each experiment, we fix the total number of training communication rounds, denoted as T, and apply it consistently to all CFL algorithms. This ensures fairness, as all CFL algorithms update their models T times. However, it is important to note that each CFL algorithm incurs different communication and computation loads per communication round. Additionally, the required resources vary depending on how quickly they converge to optimal clustering, suggesting that a fixed T might not uniformly represent fairness across all factors.

Empirically, we have observed that CFL-GP demonstrates fast convergence, which leads to reduced computation load for clustering and model updates within the given T communication rounds. Consequently, CFL-GP often incurs lower costs to achieve comparable levels of task performance. For a broader understanding of the performance improvements per iteration, we present a comprehensive comparison of these costs in Section D with detailed performance metrics in subsequent subsections.

We set $\beta_t = \frac{1}{\lfloor t/(KP) \rfloor + 1}$ for all cases. For a given fixed number of total training communication rounds T, if the clustering remains unchanged for a continuous period of $T/10$ rounds, we halt the clustering update process of CFL-GP and perform model updates using the final clustering results.

One of the key advantages of CFL-GP lies in its robustness across diverse settings without necessitating cluster-branching sensitivity-related hyperparameters, unlike other CFL algorithms such as MADMO and PACFL that depend on such hyperparameters. Instead, CFL-GP, along with IFCA, requires specifying the number of models to be learned, a decision that falls to the system engineer. In simulations, the number of clusters K is given and fixed for CFL-GP and IFCA, which may or may not be equal to the number of distinct distributions D. Interestingly, we found that CFL-GP's gradient profile matrix can be exploited to empirically determine D, which in principle could be used to choose the number of models K, e.g., $K \approx D$ if desired. Therefore, in the absence of prior knowledge or a definitive value for K, we recommend employing the methodology outlined in Appendix G to determine the optimal number of models. For algorithms that require cluster-branching sensitivity-related hyperparameters, we use the hyperparameters specified for MADMO in (Sattler, 2020)

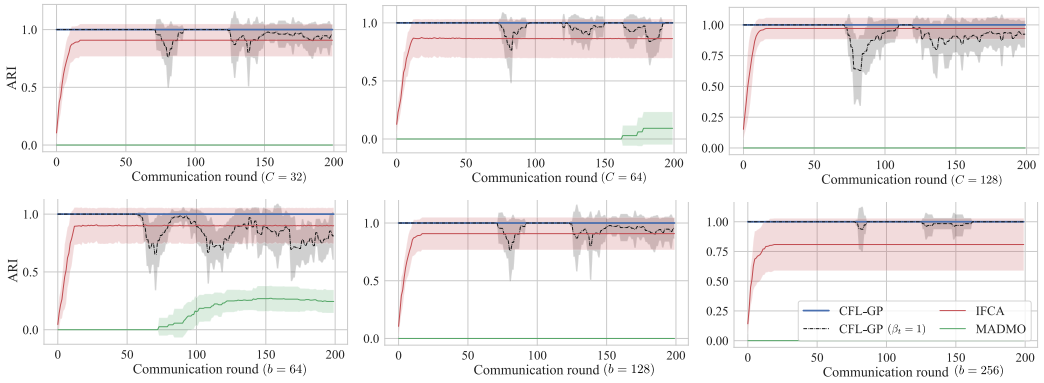

Figure 7. ARI with respect to the communication round for different numbers of clients and batch sizes, $C = 32, 64, 128$ on the top row, and $b = 64, 128, 256$ on the bottom row. CFL-GP (default setup, $\beta_t = \frac{1}{\lfloor t/(KP) \rfloor + 1}$, blue solid line) shows the *optimal clustering performance* for all the tasks with 1.0 ARI. The dashed line represents CFL-GP with $\beta_t = 1$, which relies solely on the temporal gradient gap and may exhibit clustering results that are potentially noisy.

and assess PACFL's performance using a variety of hyperparameters as detailed in Appendix F.3.

F.1. Ablation studies for feature moving average factor β

In this subsection, we show that setting the β_t to $\frac{1}{\lfloor t/(KP) \rfloor + 1}$, thereby taking a cumulative average of the gradient information, enables CFL-GP to achieve improved clustering performance. We consider the simulation environment in Section 5.2, where multiple distributions are generated for the MNIST dataset using eight different rotation transformations (0, 15, 90, 105, 180, 195, 270, 285 degrees). The given dataset is equally divided among the clients, and the CFL algorithm aims to cluster the clients into four distinct models. In evaluating the ARI, we consider optimal clustering as grouping together the closest rotation transformations to form the four clusters.

We conduct experiments with different numbers of clients and batch sizes, setting $\beta_t = 1$, and present the results in Figure 7. The first row of Figure 7 compares the ARI scores of different algorithms for a fixed batch size of $b = 128$ while varying the number of clients from 32 to 128. The second row illustrates the performance for a fixed number of 32 clients while increasing the batch size from 64 to 256.

Unlike the original setup with $\beta_t = \frac{1}{\lfloor t/(KP) \rfloor + 1}$, which consistently achieved an ARI score of 1.0 using time averaging, setting $\beta_t = 1$ introduces noise. Depending on the number of clients, specific noise patterns may not be evident. For smaller batch sizes, setting $\beta_t = 1$ leads to more noticeable noise. However, when a sufficiently large batch size is provided ($b = 256$), the noise in ARI performance becomes minimal.

Setting $\beta_t = 1$ makes CFL-GP rely solely on instantaneous gradient gaps for cluster discrimination. Consequently, if noisy gradients occur at specific time steps, they can interfere with clustering due to the inability to accumulate gradient information over extended periods. In contrast, the default setting of CFL-GP, which employs time averaging, enables the accumulation of gradient information over multiple time steps. This approach results in denoised gradient profiles, ensuring stable clustering performance.

F.2. Comparison with Personalized Federated Learning utilizing implicit clustering

One of the key differences between CFL and conventional Personalized Federated Learning (PFL) lies in the explicit client clustering performed by CFL. As mentioned in Section 1 and A, CFL proves to be highly beneficial in scenarios where the assumption of clients sharing a common representation, often presumed in the setups of PFL, is invalid or when the desired models have limited representation capability. Furthermore, CFL algorithms effectively address the challenges posed by noisy or limited client data by grouping clients together for training a single model. This approach helps prevent a decline in generalization performance that may arise from individual client fine-tuning.

Similar to CFL, certain PFL algorithms have the ability to implicitly measure client similarity and facilitate the learning of similar models among clients with comparable datasets. One of the notable and well-founded PFL algorithms in this context

is FedEM (Marfoq et al., 2021), which leverages multiple models for a given task. Each client can customize the utilization of these models based on their own weight factors. Theoretically, under certain conditions, classification problems under CFL setup can be recovered by the PFL setup defined in (Marfoq et al., 2021) (See Section 2.3 in (Marfoq et al., 2021)). In this subsection, we conduct a performance comparison between CFL and the PFL algorithm FedEM, while also evaluating the clustering performance of FedEM.

We utilize the experiment setup described in Section F.5.1, where four different rotation transformations (0, 90, 180, and 270 degrees) are applied to the MNIST datasets, resulting in distinct distribution identities. The dataset is equally divided into C clients, and each client has a heterogeneous local dataset due to the disjoint dataset division and rotation transformations (see Section F.5.1-Extensions for more details).

In this experiment, we set the number of models as four for all the algorithms. Each client in FedEM has its own model based on a weight factor with the same dimension as the number of models, indicating the extent to which they rely on each model for the classification task (See (Marfoq et al., 2021) for details). It should be noted that in this setup, FedEM ideally can recover the performance of the CFL algorithm if it adequately clusters the clients. To measure the clustering performance of FedEM, we observe the weight factors assigned to each client. If two clients have the same local data distribution, they should have the same weight factors. We consider two clients to be in the same cluster if they exhibit the highest weight for the same model. For example, Table 8 shows the weight factors of the clients after 200 rounds of learning with 20 clients in the experiment. We consider the clients c_1 and c_2 are in the same cluster as their weight factors show the largest value for the third model. In Table 9, we present the classification performance and ARI with their standard deviations for 5 different random seeds for a given number of communication rounds $T = 200$.

Table 8. Clients weight factors in FedEM ($C = 20$, $K = 4$, $T = 200$)

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
Weight factor 1	7.88e-3	1.53e-3	6.26e-9	3.10e-10	2.43e-3	0.00e+0	3.40e-3	2.38e-5	1.65e-8	1.70e-3
Weight factor 2	5.23e-3	1.15e-2	1.43e-10	1.65e-2	1.33e-2	0.00e+0	4.67e-3	1.35e-2	2.70e-2	1.02e-2
Weight factor 3	**9.87e-1**	**9.84e-1**	**1.00e+0**	**9.83e-1**	**9.84e-1**	4.13e-21	9.92e-1	**9.86e-1**	**9.73e-1**	**9.88e-1**
Weight factor 4	6.95e-8	3.31e-3	3.84e-31	4.15e-32	5.70e-32	**1.00e+0**	**1.00e-4**	4.21e-22	9.76e-24	1.94e-23

	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}	c_{17}	c_{18}	c_{19}	c_{20}
Weight factor 1	5.97e-3	8.70e-9	2.85e-39	1.26e-44	4.27e-13	2.25e-15	1.16e-3	0.00e+0	0.00e+0	6.00e-3
Weight factor 2	5.95e-3	1.98e-2	0.00e+0	1.15e-31	1.69e-2	1.66e-2	1.97e-2	0.00e+0	0.00e+0	7.92e-3
Weight factor 3	**9.87e-1**	**9.80e-1**	5.10e-18	3.70e-20	**9.83e-1**	**9.83e-1**	**9.79e-1**	8.70e-21	2.25e-18	**9.86e-1**
Weight factor 4	9.28e-4	8.94e-21	**1.00e+0**	**1.00e+0**	1.70e-25	5.87e-36	5.11e-27	**1.00e+0**	**1.00e+0**	2.69e-4

In this setup, the results presented in Table 8 and Table 9 indicate that FedEM may not capture the explicit distribution identities of the clients. This is from the observation that clients c_1 to c_5, c_6 to c_{10}, c_{11} to c_{15}, and c_{16} to c_{20}, which possess the same rotation transformation-based local datasets, do not exhibit similar weight factors as expected. The discrepancy in weight factors suggests a failure to capture the inherent similarity among clients with identical data distributions.

Furthermore, when ARI is measured using the aforementioned approach, Table 9 demonstrates that FedEM yields ARI values close to 0.0 for varying numbers of clients. Consequently, it shows lower performance compared to CFL-GP and IFCA, which achieve significantly higher ARI scores (1.0 and about 0.86, respectively), but outperforms MADMO, which fails to attain high ARI, and the Global scheme that employs a single model. Note that CFL-GP consistently shows 1.0 ARI with zero variance in all the experiments, resulting in the highest performance among the FL algorithms.

Table 9. Performance comparison between the CFL and PFL algorithms. (Classification accuracy (%) / ARI)

Algorithm	$C = 20$	$C = 40$	$C = 80$	$C = 160$
CFL-GP	**88.81±0.10/1.00±0.00**	**88.75±0.16/1.00±0.00**	**88.67±0.20/1.00±0.00**	**88.78±0.16/1.00±0.00**
IFCA	86.44±2.79/0.87±0.16	86.38±2.84/0.88±0.15	86.16±3.17/0.88±0.14	84.29±2.27/0.77±0.12
MADMO	61.90±1.42/0.02±0.07	65.64±5.23/0.20±0.27	67.41±6.54/0.30±0.26	71.53±8.01/0.50±0.28
Global	61.20±0.29/0.00±0.00	61.48±0.09/0.00±0.00	61.23±0.36/0.00±0.00	61.37±0.37/0.00±0.00
FedEM	76.94±3.47/0.00±0.01	76.57±3.49/0.00±0.01	76.42±3.21/0.00±0.00	76.14±3.30/0.00±0.01

F.3. Comparison with recent CFL algorithms from different communication protocols

In traditional Federated Learning (FL) systems, the standard communication protocol involves transmitting gradient information from multiple clients to a central unit without sharing the raw local data. This method provides a moderate level of data privacy protection. CFL-GP, IFCA, MADMO, and the Global Model (FedAvg) follow a comparable communication

protocol, where clients transmit gradient information related to a set of models (CFL-GP, IFCA, MADMO, and the Global Model) or model evaluation results (IFCA).

Principal Angles analysis for Clustered Federated Learning (PACFL). Recently, Vahidian et al. (Vahidian et al., 2023) proposed a CFL algorithm with a communication protocol that permits clients to send principal vectors of their local datasets to a central unit for clustering. Such an approach expands the algorithm's design scope for clustering, as it allows the central unit to access a wider array of information. This includes both the transformed data (in the form of principal vectors) and traditional gradient information. The incorporation of this additional data type endows PACFL with an enhanced level of flexibility in feature design, and in certain scenarios, PACFL outperforms ICFA and MADMO (Vahidian et al., 2023).

For cases with the potential heightened privacy concerns inherent in transmitting the principal vectors of local datasets, the authors recommend the integration of additional privacy safeguards in privacy-sensitive scenarios, e.g., the use of encryption methods and differential privacy techniques.

An additional key aspect of PACFL is its utilization of a clustering threshold parameter, ν (corresponding to β in (Vahidian et al., 2023)). A higher value of ν results in reduced clustering sensitivity, thereby favoring the generation of more globalized models with fewer distinct models. Conversely, a lower ν increases clustering sensitivity, leading to the creation of more individualized models for each client. This parameter plays a role in balancing the need for model specificity against the desire for broader applicability across various clients.

We implemented PACFL with various ν and compared its performance on the same simulation environment used in Appendix F.2. In Figure 8, we present the achieved accuracy and ARI of PACFL relative to the number of clients, alongside

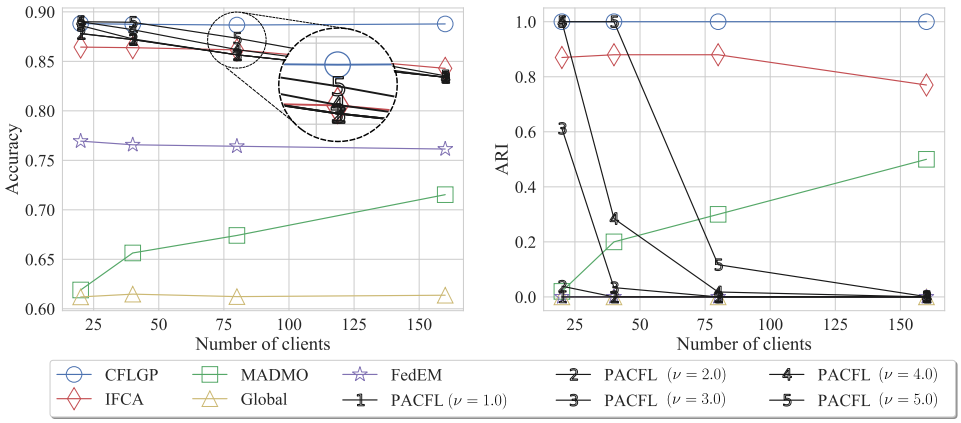

Figure 8. Achieved accuracy and ARI according to the number of clients.

performance measures for CFL-GP, IFCA, MADMO, Global Model, and FedEM, adopted from Appendix F.2. The results demonstrate that PACFL achieves optimal performance with fewer than 50 clients, particularly when ν is set to 4 or 5. However, as the number of clients increases, PACFL's ARI decreases, indicating reduced effectiveness of principal angle-based clustering with sparse client data. The simulation involves equally partitioning the dataset among clients and applying rotational transformations, leading to fewer samples per client as the number of clients increases. This trend underscores the consistent performance of CFL-GP, which, relying solely on gradient information, achieves a stable ARI of 1.0. This highlights the critical role of gradient data in effective clustering.

CFL-GP operates independently of hyperparameters that affect clustering sensitivity, thereby matching the peak performance of PACFL. While PACFL may require hyperparameter adjustments to achieve an ARI of 1.0, CFL-GP consistently delivers comparable outcomes without such tuning. Instead, CFL-GP relies on a predetermined number of models. This number can be determined through a gradient-based method, detailed in Appendix G, which accurately estimates the appropriate model count.

F.4. Experiments on Multiple Linear Regression Tasks

Through this subsection, we provide a more detailed configuration for the experiments in Section 5.1.

We consider a CFL problem where the objective is to learn three linear models for regression tasks. We have three different data distributions, $D = 3$, and set three models for the problem, $K = 3$. For a given $k \in [3]$ and ϕ_k, we generate data pairs $\mathbf{x} = (x, y)$ as $y = x\tan(\phi_k) + n$ where $n \sim \mathcal{N}(0, 0.2^2)$ and x follows $\mathcal{U}(0, \cos(\phi_k))$, a uniform distribution over $[0, \cos(\phi_k)]$ as shown in Figure 9. We consider three angles (ϕ_1, ϕ_2, ϕ_3) = ($\Delta\phi$, 0, $-\Delta\phi$) for vaious $\Delta\phi$s.

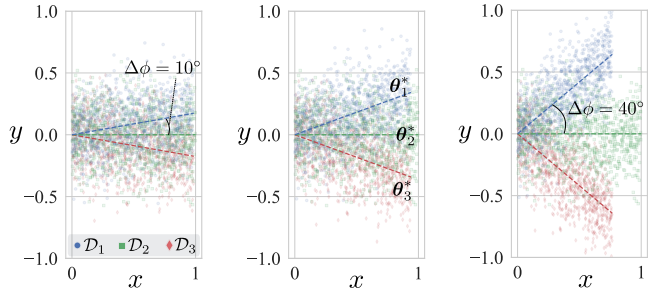

Figure 9. Multiple linear regressions. The data distributions depicted in the subplots of Figure 9 correspond to multiple linear regression tasks with $\Delta\phi$ values of 10, 20, and 40 degrees, from left to right. The dashed lines in each subplot represent the output of the optimal models for the respective data distributions in the regression tasks.

The first (left) subplot in Figure 9 depicts the scenario where $\Delta\phi = 10°$. The blue circular markers correspond to the first data distribution $\mathcal{D}_1$ of $\phi_1 = \Delta\phi$, the green square data markers represent the second distribution $\mathcal{D}_2$ of $\phi_1 = 0$, and the diamond-shaped data markers denote the third distribution $\mathcal{D}_3$ of $\phi_1 = -\Delta\phi$. Each client is assigned a specific distribution identity from this set. As the value of the cluster gap $\Delta\phi$ decreases, the CFL algorithms may face a growing challenge in distinguishing between similar data distributions. This is because the distance between the optimal models becomes smaller, and the gradients from different distributions tend to become more similar.

The k-th linear model is characterized by its parameters $\boldsymbol{\theta}_{k,1}$ and $\boldsymbol{\theta}_{k,2}$. For a given data point x, the model's output, denoted as $\hat{y}$, is computed as $\hat{y} = \boldsymbol{\theta}_{k,1}x + \boldsymbol{\theta}_{k,2}$ For each k, the weights are initialized using one of two methods: (i) $\boldsymbol{\theta}_{k,1}^{(0)} \sim \mathcal{U}(-0.8, 0.8)$, or (ii) $\boldsymbol{\theta}_{k,1}^{(0)} \sim \mathcal{U}(-1.6, 1.6)$. In both cases, the bias term $\boldsymbol{\theta}_{k,2}^{(0)}$ is set to 0.

The dashed line plots shown in the three subplots of Figure 9 depict the output of the optimal models that can minimize the Mean Squared Error (MSE) for each respective distribution. Introducing a label standard deviation of 0.2, the CFL algorithm would achieve an optimal performance MSE of 0.04 if it could attain an optimal clustering.

Configuration. CFL-GP and IFCA require a number of models prior to learning. We set $K = 3$, $T = 200$, and learning rate to 0.1. MADMO requires branching parameters, `EPS_1` and `EPS_2`, that lead to recursive bipartitioning of the client sets. We use `EPS_1` and `EPS_2` as 0.02 and 0.6, respectively (See (Sattler, 2020)).

F.5. Experiments on MNIST Dataset

For the simulations in Section 5.2, we adopt a CFL problem setup from (Ghosh et al., 2020). In this setup, the MNIST dataset is partitioned among multiple clients, with each subset undergoing a rotation transformation. This configuration allows us to evaluate the clustering accuracy of CFL algorithms by assessing their ability to group clients based on these transformations.

To enhance the complexity of the experiments, we extend the simulation environment by introducing eight different rotation angles: 0, 15, 90, 105, 180, 185, 270, and 285 degrees. A subset of the results from this simulation is presented in Section 5. Here, we provide a comprehensive description of the experimental details.

The MNIST dataset is divided into eight disjoint parts, with each part being subjected to one of the rotation transformations mentioned above. These transformed datasets are then equally distributed among a total of C clients, with each client receiving a local dataset that corresponds to one of the eight rotational transformations. Therefore, each group of $C/8$ clients has the same rotation transformation, but they also exhibit heterogeneity as the MNIST dataset is divided disjointly among the clients. The 70% of each local dataset is used as a training dataset for the corresponding client, and the remaining 30% is used as a test dataset.

The parameterized model consists of two dense layers: the first layer receives a 28×28 dimension with the 1-channel image

as an input and has a hidden dimension size of 200. The second dense layer receives the output from the first layer, which is a vector of size 10, where 10 is the number of distinct labels in the MNIST dataset. The output from the second layer is then used to predict labels through the log softmax activation. The cross-entropy loss is used.

To assess the robustness of the algorithms, performance measurements are carried out across a range of environments with varying numbers of clients, starting from 8 and increasing to 128. For all the algorithms, we have a default setup as $b = 64$, $C = 32$, and we set the learning rate to 0.1, $P = 2$, and $T = 200$.

Configuration. MADMO primarily employs a model averaging protocol. However, to ensure fairness with other algorithms, we implement a gradient averaging protocol by limiting the number of local updates per communication round to one. We follow the hyperparameter configuration of MADMO as described in (Sattler et al., 2020). For IFCA, the gradient averaging version algorithm [Algorithm 1, option I](Ghosh et al., 2020) is used. The Global Model scheme is implemented by setting $K = 1$ in CFL-GP, which means that the learning agent will train a single global model, and no client partitioning is performed.

F.5.1. Additional Simulation Results

Classification accuracy and ARI vs. communication round. We present additional plots illustrating the ARI and Accuracy according to the communication round t. Each subplot in Figure 10 is labeled with the specific batch size b and the number of clients C used in the experiments. The shaded region in the plots represents the standard deviation of the performance, calculated based on 10 repeated experiments, with the average performance serving as the line plots.

Among the CFL algorithms, CFL-GP achieves optimal clustering in a *single round*, except when the number of clients is 8. As a result, CFL-GP demonstrates enhanced capability to update the models for each distribution accurately, leading to higher classification accuracy. This stands in stark contrast to the competing algorithms, which often fail to converge or achieve optimal clustering even after 200 rounds. These competing algorithms exhibit suboptimal performance throughout the simulation.

Moreover, baseline algorithms display distinct ARI performance depending on the number of clients, consequently influencing the task performance. In contrast, CFL-GP exhibits robustness even in scenarios with a small number of clients ($C = 8$), achieving optimal clustering within just 20 rounds.

Figure 11 presents the results obtained when considering various batch sizes with a fixed number of 32 clients. Except for the batch size of 32, CFL-GP consistently achieves optimal clustering in just one round. Even with a batch size of 32, optimal clustering is achieved after approximately 80 rounds. On the other hand, IFCA and MADMO fail to achieve optimal clustering in these experiments. Additionally, MADMO fails to detect any clusters when the batch sizes are set to 128 and 256, performing equivalently to using a global model. CFL-GP consistently attains the highest ARI in all experiments, resulting in superior classification accuracy.

Performance comparison with a different number of models. Based on the main results provided in Section 5.2, we run CFL-GP and IFCA on the different number of models $K \in \{4, 8\}$. Note that the number of initial models is obtained by using the method that we propose in Appendix G, which is the singular vector analysis utilizing a gradient profile matrix.

Table 10. Performance comparison: Classification accuracy (%)

Algorithm	$C = 8$	$C = 16$	$C = 32$	$C = 64$	$C = 128$
CFL-GP (K=4)	86.8±0.35	87.2±0.21	**87.2±0.18**	**87.2±0.28**	**87.3±0.24**
CFL-GP (K=8)	**87.7±0.22**	**87.6±0.33**	86.7±1.22	**87.2±0.26**	87.1±0.19
IFCA (K=4)	79.3±2.62	82.8±3.18	85.2±2.85	85.4±2.58	86.0±2.28
IFCA (K=8)	84.1±3.45	82.0±3.57	85.0±3.17	**87.2±0.06**	**87.3±0.10**
MADMO	63.7±4.02	69.6±2.76	72.5±2.17	76.4±2.42	81.8±2.20
Global	58.0±0.45	58.3±0.45	58.3±0.37	58.3±0.51	58.4±0.38

Table 10 outlines the classification accuracy results across various client counts, ranging from 8 to 128. The results indicate that with the number of models $K = 4$ and $K = 8$, determined using the method described in Appendix G, CFL-GP can appropriately cluster the clients and achieve high performance. Similarly, IFCA, which requires a predefined number of models, also demonstrates high performance, surpassing other baseline methodologies when using the estimated number of models.

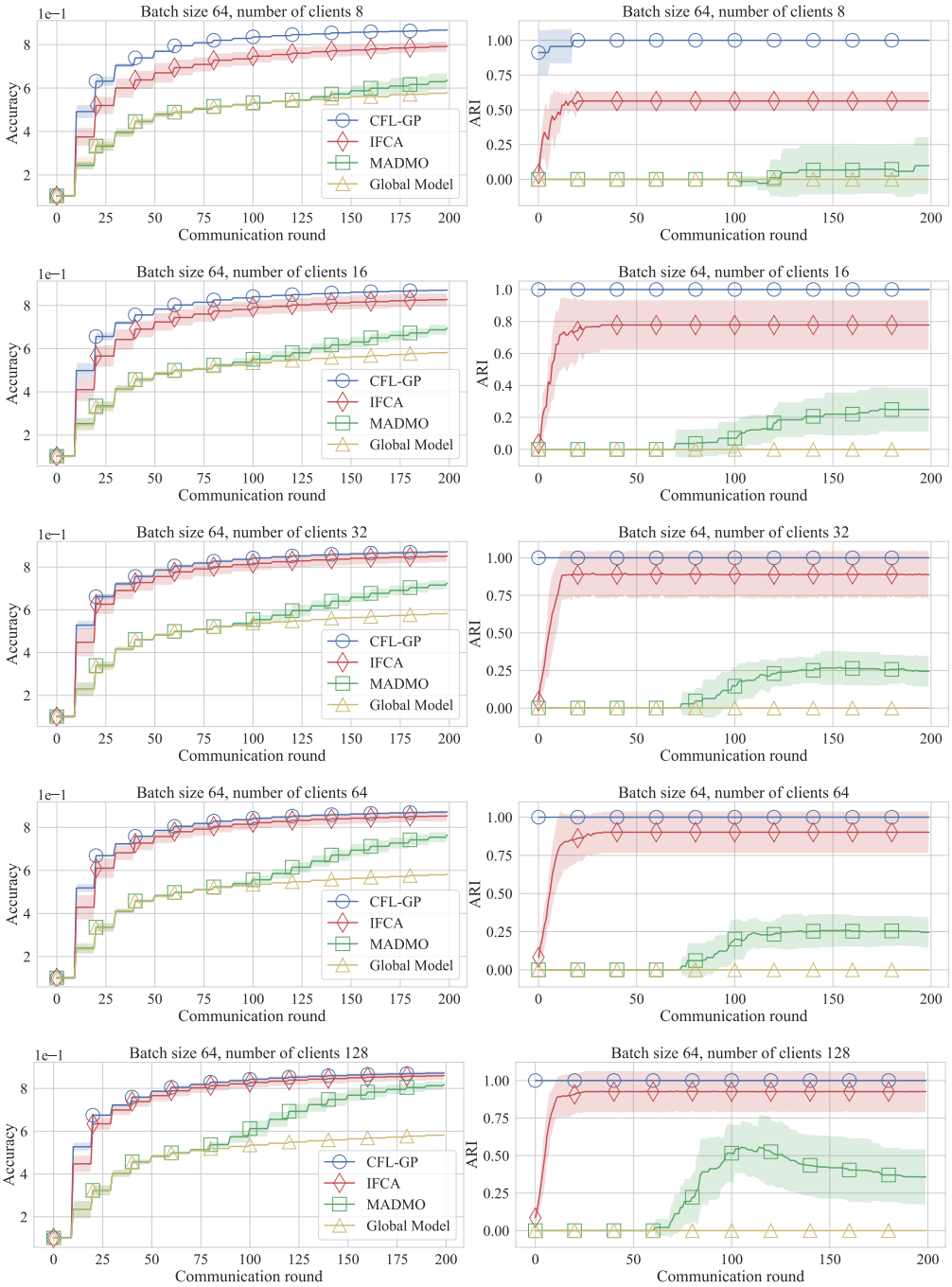

Figure 10. Classification accuracy and ARI according to communication round. CFL-GP achieved *optimal clustering* within 20 rounds in all experiments. In contrast, other CFL algorithms failed to achieve optimal clustering within 200 rounds, exhibiting varied ARI performance depending on the number of clients.

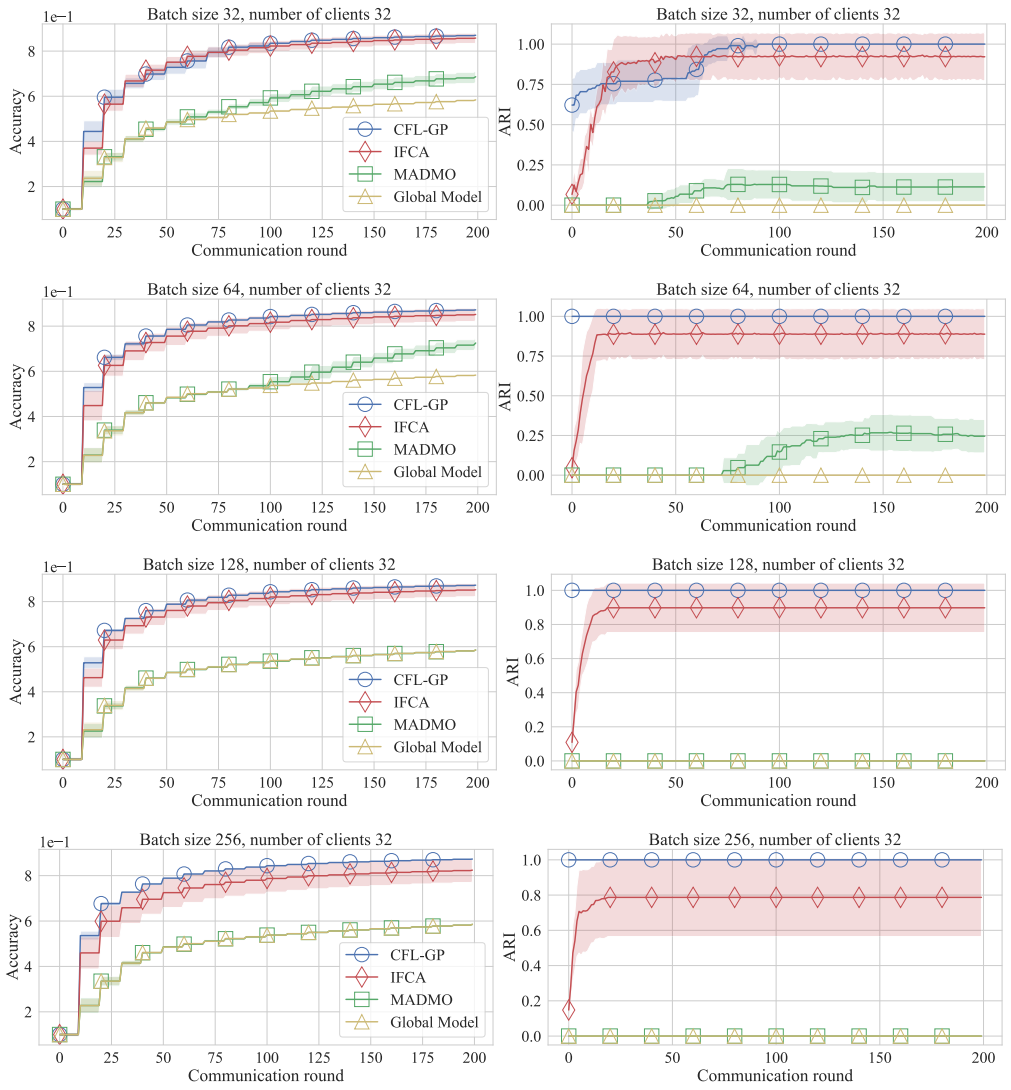

Figure 11. Classification accuracy and ARI trends across communication rounds. CFL-GP achieved *optimal clustering* within 90 rounds in all experiments. In contrast, other CFL algorithms failed to achieve optimal clustering, exhibiting varied ARI performance depending on the batch size.

Visualization of Gradient Profile Matrix. In Figure 12, we visualize the projected features of each client, $\boldsymbol{g}'_c \in \mathbb{R}^K$ obtained during the spectral clustering based on the gradient profile matrix. The first row displays results for a batch size of 64 with 64 clients, while the second row shows results for 128 clients. Each column corresponds to a specific time step, specifically $t = 1$, 5, 9, and 13.

All the features are further randomly projected into a 3-dimensional space for visualization purposes. Figure 12 illustrates that as t increases, accumulating more gradient information, the distances between the sets of reduced features for each cluster become larger. Notably, the scatter of each set (e.g., the clients with Rotation Transformation (RT) 0 and 15 degrees) at $t = 1$ decreases progressively with increasing t. CFL-GP leverages these features for clustering, effectively identifying the four clusters. In these experiments, CFL-GP achieves optimal ARI in just one round, contrasting with IFCA and MADMO, which attain ARI scores of approximately 0.9 and below 0.5, respectively.

Additional simulation environments. In addition to the results presented in the main text, we adopt the same experimental configuration from (Ghosh et al., 2020) where the MNIST dataset is divided into four disjoint parts instead of eight, each undergoing rotational transformations at angles of 0, 90, 180, and 270 degrees. In this environment, the default batch size is

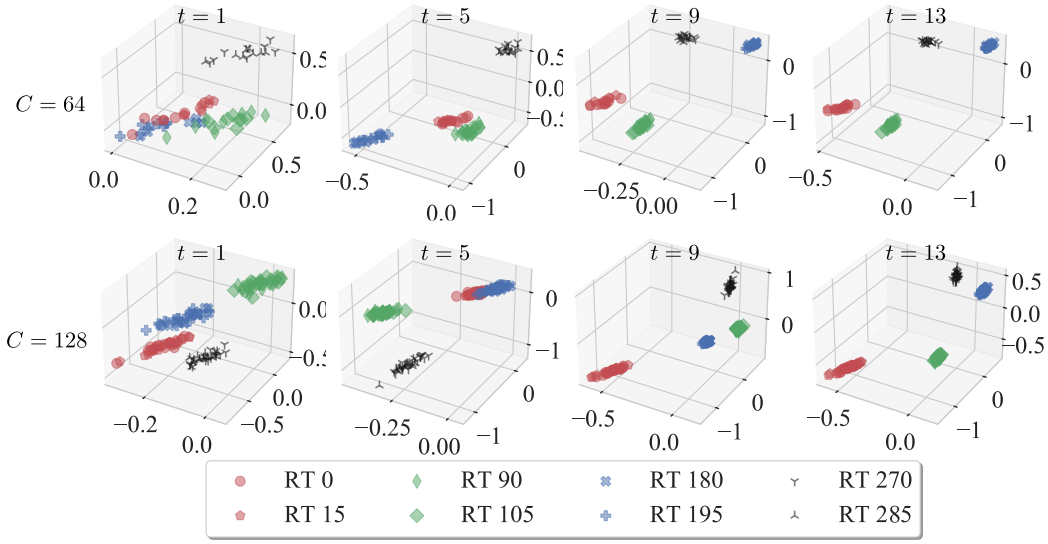

Figure 12. The visualization of reduced client features obtained at different t, $t = 1, 5, 9$, and 13 for two scenarios, where the number of clients is 64 and 128, respectively. Each data point represents a reduced client feature, $\boldsymbol{g}'_c$. Notably, CFL-GP achieves optimal clustering (ARI of 1.0) within a single clustering round $t = 1$ using these features.

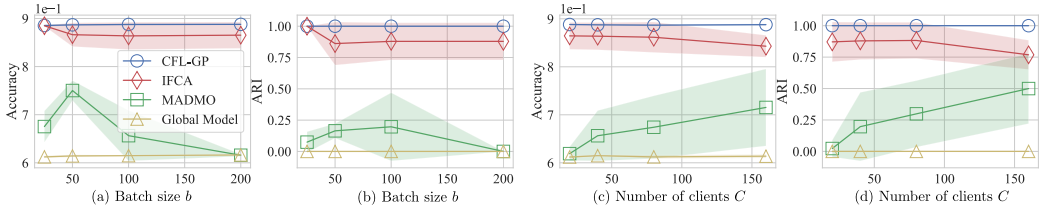

Figure 13. Rotated MNIST setup with the four different rotation transformations, 0, 90, 180, and 270 degrees. Accuracy and ARI are plotted against batch size and number of clients in subplots (a)-(d), respectively. CFL-GP achieves 1.0 ARI in all the environments, resulting in the highest classification accuracy.

set to $b = 100$, and the number of clients is set to $C = 40$.

Figure 13 depicts the classification accuracy and ARI based on different batch sizes and numbers of clients. CFL-GP consistently achieves an ARI of 1.0 for all batch sizes and client numbers, indicating optimal clustering performance. Consequently, CFL-GP also consistently exhibits the highest accuracy values among the FL algorithms. In contrast, other CFL algorithms exhibit sensitivity to batch sizes and the number of clients, leading to significant variations in ARI and task performance. The global model, which utilizes a single model for all experiments, consistently shows the lowest accuracy across all scenarios.

F.5.2. Configuration for experiments on the mixture of distributions

We detail the simulation setup for the results in Figure 3. To evaluate the robustness of CFL algorithms against distribution similarity, the MNIST dataset is partitioned into three distinct subsets. The first subset undergoes a rotation transformation of 0 degrees, while the second subset undergoes a rotation transformation of 180 degrees. For the third subset, we randomly assign each sample to one of the two rotation transformations, with probabilities κ and $1 - \kappa$, respectively, creating a mixture distribution. Each subset is then equally divided among four clients. We set $T = 400$ and $\kappa \in \{0.1, , 0.2, \ldots, 0.9\}$. Throughout the training, the clustering process is active, with $T_{\text{cl}} = T$.

F.5.3. Extension: Integration with a different dataset

To evaluate the effectiveness of CFL algorithms in distinguishing between datasets, we created a hybrid dataset by merging MNIST and Fashion-MNIST. This dataset was distributed among clients, with configurations of 32, 64, and 128 clients.

Following the primary simulation parameters detailed in Appendix F.5, we maintained consistent learning hyperparameters and CFL baseline settings. Our experimental protocol included a batch size of 256 and spanned 200 communication rounds. Clients assigned portions of the MNIST dataset underwent eight distinct rotational transformations, as outlined in Appendix F.5. We assessed performance using classification accuracy (ACC) and the Adjusted Rand Index (ARI), conducting five rounds of simulation with different random seeds. The results, summarized in Table 11, present the average ACC/ARI metrics and their respective standard deviations.

Table 11. Performance Comparison: Accuracy (%) and ARI for MNIST + Fashion-MNIST Dataset

Algorithm	$C = 32$ (ACC/ARI)	$C = 64$ (ACC/ARI)	$C = 128$ (ACC/ARI)
CFL-GP	**82.4±0.03/1.00±0.00**	**82.8±0.30/1.00±0.00**	**82.8±0.18/1.00±0.00**
IFCA	80.4±1.12/0.95±0.04	78.2±1.80/0.84±0.12	81.1±1.15/0.95±0.04
MADMO	61.5±1.68/0.07±0.10	67.9±5.17/0.36±0.28	68.8±5.43/0.59±0.16
Global Model	60.5±0.27/0.00±0.00	60.8±0.23/0.00±0.00	60.6±0.20/0.00±0.00

CFL-GP consistently outperformed baseline models, showcasing superior task performance and achieving up to a 21% increase in accuracy. IFCA also displayed comparable performance to CFL-GP. The Global Model approach significantly underperformed in this scenario, as it is unable to separate the two distinct data distributions and the limited capability of the multi-layer perceptron models to achieve high accuracy across combined datasets. Similarly, MADMO exhibited lower performance, with its ARI lagging behind CFL-GP and IFCA.

F.6. Experiments on Wireless Channel Datasets

In this section, we provide a detailed description of the simulation setup corresponding to **[Exp 1: Deep AE, COST2100]** in Table 1 and offer additional analysis based on the results.

We consider a wireless Channel State Information (CSI) compression problem, which is a crucial challenge in wireless communication, particularly for enhancing efficiency in next-generation systems (Lin, 2022; Wen et al., 2018; Wang et al., 2018). This problem parallels the well-studied field of neural network-based image compression (Jiang, 1999; Ma et al., 2019; Hu et al., 2021b). Below, we outline the setup for this problem.

Channel State Information and Deep Autoencoder-based compression. In wireless communications, channel state information (CSI) refers to various types of information regarding a communication link between a transmitter (Tx) and a receiver (Rx), and CSI not only changes in real-time but also varies depending mainly on the location of the Tx and Rx. If the Rx compresses CSI more efficiently and sends it to the Tx, the Tx can design a more robust signal and achieve higher data transmission efficiency.

Autoencoder-based CSI compression methods (Wen et al., 2018; Wang et al., 2018) have consistently outperformed traditional compression techniques, including compressive sensing methods. This approach is regarded as a promising technology with vast potential for application beyond 5G and next-generation communication systems by effectively mitigating signal interference and optimizing communication efficiency (Lin, 2022).

Application of CFL to wireless communication systems. Despite the potential of autoencoder-based CSI compressors, several critical challenges need to be addressed for practical deployment: (a) Channel Distribution Heterogeneity: This challenge arises due to varying channel distributions across clients, where a client is defined as a data center and a set of terminal devices (or a base station - a set of mobile users pair). Environmental factors significantly influence the communication link between terminal devices and the base station. (b) Limited Representation Capability: The autoencoder structure compresses original data into a codeword, i.e., a latent vector having a limited size. Due to the heterogeneity of the channels, using a single model for all distributions may not be effective, as representing diverse heterogeneous distributions within a fixed-size latent vector leads to higher distortion between input and recovered output.

To address these challenges, we employ a CFL scenario where algorithms are encouraged to cluster clients based on channel heterogeneity or similarity, and train a distinct deep autoencoder for each cluster. This method allows each cluster to use a tailored autoencoder, resulting in more efficient compression compared to a single autoencoder handling all channel distributions within a limited latent space. This approach effectively mitigates the identified issues and facilitates the practical deployment of tailored models.

Datasets and Autoencoder Models. We utilize the COST2100 (Wen et al., 2018) channel model dataset, which is one of the most widely used benchmark datasets for deep learning-based CSI compression tasks. The COST2100 dataset includes two classes of CSI; the first class is an indoor type CSI $\mathcal{D}_1$, characterized by a weak scattering pattern and a small delay component, and the second class is an outdoor type CSI, characterized by a severe scattering pattern and a relatively high delay component, $\mathcal{D}_2$. Each CSI sample is a 32×32 size matrix, and each element is a complex value. Each sample can then be expressed by 2,048 floating point numbers. The entire dataset contains 200,000 samples, and we divide them into C clients. Then, half of the C clients possess local datasets corresponding to the COST2100 indoor dataset, and the other half comprises COST2100 outdoor datasets. The CFL algorithms are encouraged to group clients according to their dataset identities and, ideally, learn two distinct autoencoders that share an encoder. We denote this simulation environment as CD1 (Channel Dataset 1).

In addition, we create an additional channel model dataset by utilizing the channel model generator (Jaeckel et al., 2021). This dataset adheres to the established industrial standard model, specifically the 3GPP 38.901 channel model. We generate five distinct wireless channel distributions, denoted as $\mathcal{D}_1, ..., \mathcal{D}_5$, and extract 20,000 instances of CSI from each distribution as detailed in (Kim, 2024). Similar to the COST2100 dataset, each CSI sample in the dataset is represented as a 32×32 matrix of complex values, resulting in a representation of 2,048 floating-point numbers. This dataset with the simulation setup is referred to as CD2.

We adopt Inception block (Szegedy et al., 2015) and residual connection (He et al., 2016) to build deep autoencoders, which are widely used for CSI compression (Wen et al., 2018; Lu et al., 2020).

The encoder module comprises three blocks, each with specific convolution layers and associated parameter sets. In the first block, there are three convolution layers with parameter sets (2,2,3,3), (2,2,1,9), and (2,2,9,1), respectively. Here, the first input represents the input data channel dimensions, and the second input represents the output data channel dimensions. The third and fourth inputs specify the width and height of the kernel, respectively. All convolution layers in this block are connected with batch normalization and leaky ReLU activation. The second block consists of a single convolution layer with a parameter set of (2,2,3,3). It is also connected with a batch normalization layer. The third block contains three convolution layers with parameter sets (2,2,3,3), (2,2,1,3), and (2,2,3,1), respectively. Similar to the previous blocks, these convolution layers are connected with batch normalization and leaky ReLU activation. The outputs from these blocks are combined based on the channel dimension. Subsequently, an additional refinement step is performed using a convolution layer with a parameter set of (6,2,1,1). The resulting output is then vectorized. Finally, a dense layer is utilized to generate the compressed output, which is determined by the desired compression ratio.

The compression ratios are set to 2 and 128 for the CD1 and CD2, respectively. The autoencoder models have approximately 4.2M and 72K parameters. The latent vector sizes, N_{cl}, are $N_{\text{cl}} = 1024$ and $N_{\text{cl}} = 16$ for the two models, respectively.

The decoder module in the system takes a vector of size N_{cl} and performs a series of operations to reconstruct the image. First, the vector is passed through a dense layer with 2048 outputs. The resulting output is then reshaped to match the original image shape. After reshaping, a convolution layer with a parameter set (2,2,5,5) is applied to process the data, along with batch normalization and a leaky ReLU activation layer.

To reconstruct the image, a set of residual blocks is repeated. Each block consists of convolution layers with channel dimensions related to a given parameter N_{dc}. The input to the blocks is split into two sub-blocks. The first sub-block consists of three convolution layers with parameter sets (2,N_{dc},3,3), (N_{dc},N_{dc},1,9), and (N_{dc},9,1). The second sub-block consists of two convolution layers with parameter sets (2,N_{dc},1,5) and (N_{dc},N_{dc},5,1). The outputs from these sub-blocks are combined along the channel dimension and passed through a convolution layer with a parameter set (2N_{dc},2,1,1).

The processing of each block is completed by adding its unprocessed input to the output. This block processing is repeated three times, with N_{dc} values of 9, 7, and 5, respectively. Finally, the original input is estimated using a leaky ReLU activation function.

All the leaky ReLU activations have a slope of 0.3. Xavier initialization is applied to all the weights in the model. For the batch normalization layer, the weight parameter is initialized as one, and the bias parameter is initialized as zero.

Configuration. The model averaging protocol is used for all the algorithms. Specifically, for CFL-GP, we employ the model averaging procedure described in Algorithm 4. Each client performs a local update for one epoch during each communication round. This means that each client divides its local dataset into multiple mini-batches of size 200 and uses them to update the model received from the central unit (CU). If the number of clients is smaller, each client performs a

larger number of local updates during each communication round since they possess a larger amount of data. The branching parameters for MADMO are set to the values specified in the MADMO codebase (Sattler, 2020). The implementation of FEDAVG follows the methodology described in (McMahan et al., 2017).

In the simulation corresponding to CD1, CFL-GP employs gradient compression with a ratio of 100 to manage the gradient profile matrix, i.e., $d/100$ components are randomly selected from a gradient vector to update the gradient profile matrix. This gradient compression enables CFL-GP to achieve effective clustering with the compressed information while preserving clustering optimality. For more detailed information, see Appendix E.

We use the MSE loss, the Adam optimizer (Kingma & Ba, 2015) for the local update process of the clients with a 0.002 learning rate and 200-size minibatch. We also use cosine annealing with warm restarts (Loshchilov & Hutter, 2016) for scheduling the learning rate during training. Specifically, the learning rate is reset to its original value after every 125,000 mini-batch updates by each client. Each client conducts a local update of 1 epoch during each communication round as in F.8. We use $T=2000$, $P=5$ and $T=1000$, $P=2$ for CD1 and CD2, respectively.

During the model update process, all clients share the weights of the encoder part of the autoencoder. After updating the models, we collect the encoder weights from each model and compute their average. Specifically, after Line 5 in Algorithm 4, the computation is performed as follows: $\boldsymbol{\theta}_{k,\text{ENC}}^{(t+1)} = \frac{1}{C}\sum_{c\in C}\boldsymbol{\theta}_{k_c^{(t)},\text{ENC}}^{(t+1)}$. Here, $\boldsymbol{\theta}_{k,\text{ENC}}^{(t)}$ represents the set of weights in the encoder of model $\boldsymbol{\theta}_k^{(t)}$. This weight averaging ensures that the encoder weights of all models have the same values after each update.

F.6.1. ADDITIONAL SIMULATION RESULTS

In this subsection, we observe the compression performance and clustering performance. The compression performance is measured using the normalized mean square error, which is calculated by dividing the squared Frobenius norm of the difference between the input and output data instances by the squared Frobenius norm of the original input data instance. A lower normalized mean square error indicates a better compression performance, as it means the input and output are more similar element-wise. The clustering performance is measured using the ARI.

Robustness of CFL-GP over practical scenarios. In Figure 14, the results from the simulation with COST2100 dataset are shown for 16 and 32 clients (data centers). The vertical axis on the left represents the normalized MSE in dB scale. In Figure 14, the right column demonstrates that CFL-GP achieves optimal clustering in the first round itself. As a result, CFL-GP efficiently trains autoencoders by grouping clients with similar channel distributions into one cluster, leading to the lowest MSE values, as observed in the left column. Notably, when comparing the point at which optimal ARI is achieved, it is in stark contrast that MADMO takes over 1000 rounds to achieve meaningful ARI. MADMO requires significant computational resources for bipartitioning parameter calculation until clustering is completed. In contrast, CFL-GP saves considerable resources by converging to optimal clustering in practical scenarios. In particular, it is observed that IFCA may not discover any effective clusters.

In Figure 15, the results obtained from the simulation environment CD2 are provided. In this experimental setup, CFL-GP required 130, 250, and 750 rounds to achieve ARI of 1.0 for 10, 20, and 40 clients, respectively. Other algorithms failed to achieve ARI greater than 0.55. In this scenario, CFL-GP requires a relatively high number of communication rounds to achieve a stable optimal ARI compared to other experiments, which may result in higher computation and communication costs than the baselines. However, CFL-GP ultimately achieves a high ARI, and the results demonstrate a compression performance improvement of up to 2dB compared to competing CFL algorithms.

In Figure 16, we present the client features obtained from the simulation with CD2 when C = 40. The four subplots represent the results at t = 1, 160, 320, and 480, respectively, where each subplot shows the reduced feature vectors of the clients, $\boldsymbol{g}_c' \in \mathbb{R}^K$. Given the presence of five distinct distribution identities, the reduced feature vectors are 5-dimensional. For visualization, we randomly projected these vectors onto a 3-dimensional space.

Initially, in the first round, the differences between the feature vector sets of each cluster are not clearly discernible, except for the sets corresponding to the blue and green clusters ($\mathcal{D}_1$ and $\mathcal{D}_3$). However, as CFL-GP accumulates more gradient information, by t = 480, the feature vectors begin to form distinct sets based on distribution identity, achieving an ARI of 1.0. In contrast, IFCA and MADMO exhibit lower performance compared to CFL-GP, with correspondingly lower ARI.

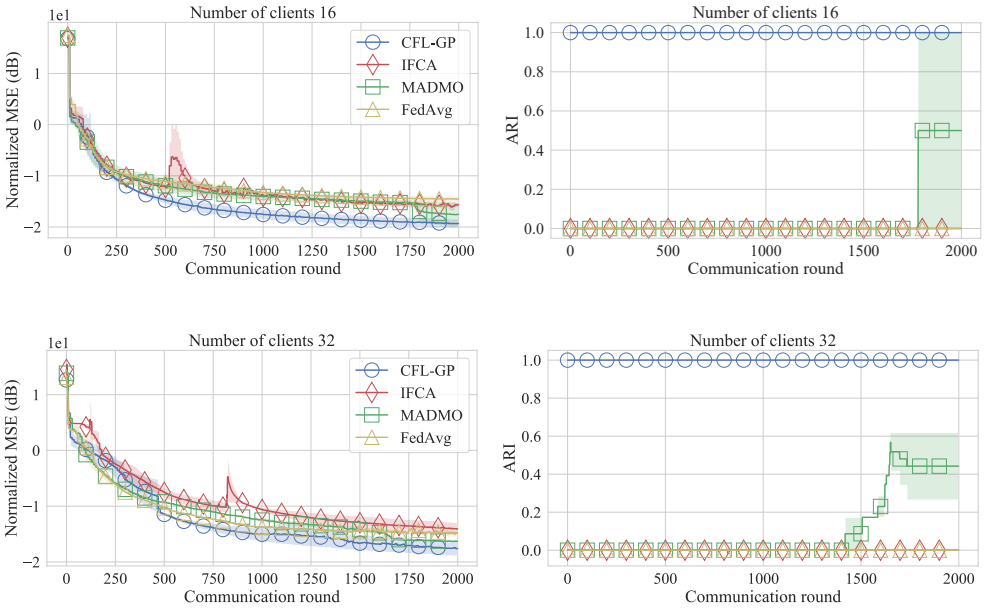

Figure 14. [Deep autoencoder, COST2100 dataset (CD1)] Normalized MSE and ARI achieved by each CFL algorithm according to communication round. CFL-GP achieves optimal clustering in the first round, resulting in the lowest normalized MSE.

F.7. Experiments on CIFAR10 and Resnet18

In this subsection, we provide a detailed description of the simulation setup for **[Exp 2: ResNet18, CIFAR-10]** in Table 1 and offer an additional analysis of the results obtained.

Our simulations utilize the CIFAR-10 dataset (Krizhevsky et al., 2009), which comprises 60,000 32x32 3-channel images across 10 classes. The dataset is randomly partitioned among the clients, denoted as C. Each client is assigned a ResNet18 model (He et al., 2016), a widely used convolutional neural network architecture.

To introduce label heterogeneity, we permute the labels for half of the C clients. Specifically, for these clients, we modify the label values by adding 5 and taking the modulus 10, resulting in a permutation of the original labels. This creates a scenario where the same class of images has different label values among clients. Consequently, CFL algorithms must detect and cluster clients based on label heterogeneity, thereby enabling the learning of distinct classification models for each cluster.

To facilitate efficient sharing of feature extraction capabilities among clients, we employ a technique known as weight sharing. Specifically, we customize only the final dense layer of the ResNet18 model while the remaining layers are shared across all clusters. This approach allows clients to train the shared feature extraction modules while using distinct classifiers to process these features. The shared weights are updated using gradient information aggregated from all clients. The concept of weight sharing with CFL, introduced in (Ghosh et al., 2020), has demonstrated effectiveness across various tasks. Neural network updates are performed using the cross-entropy loss function.

Configuration. As mentioned, client features $\boldsymbol{g}_c$ are constructed based solely on the gradients of the last layer of ResNet18 for CFL-GP. We utilize the model averaging protocol, maintaining the same configuration for local updates as in F.6. Each client performs one epoch of training during each communication round, using the Adam optimizer with a learning rate of 0.01 for the local updates. $T = 300$, $b = 100$ and $P = 2$, with varying values of $C = 20$, 40, and 80.

F.7.1. Additional Simulation Results

We observe the classification accuracy and ARI trends across communication rounds corresponding to Table 1. Figure 17 presents the accuracy and ARI achieved by each CFL algorithm under the specified conditions.

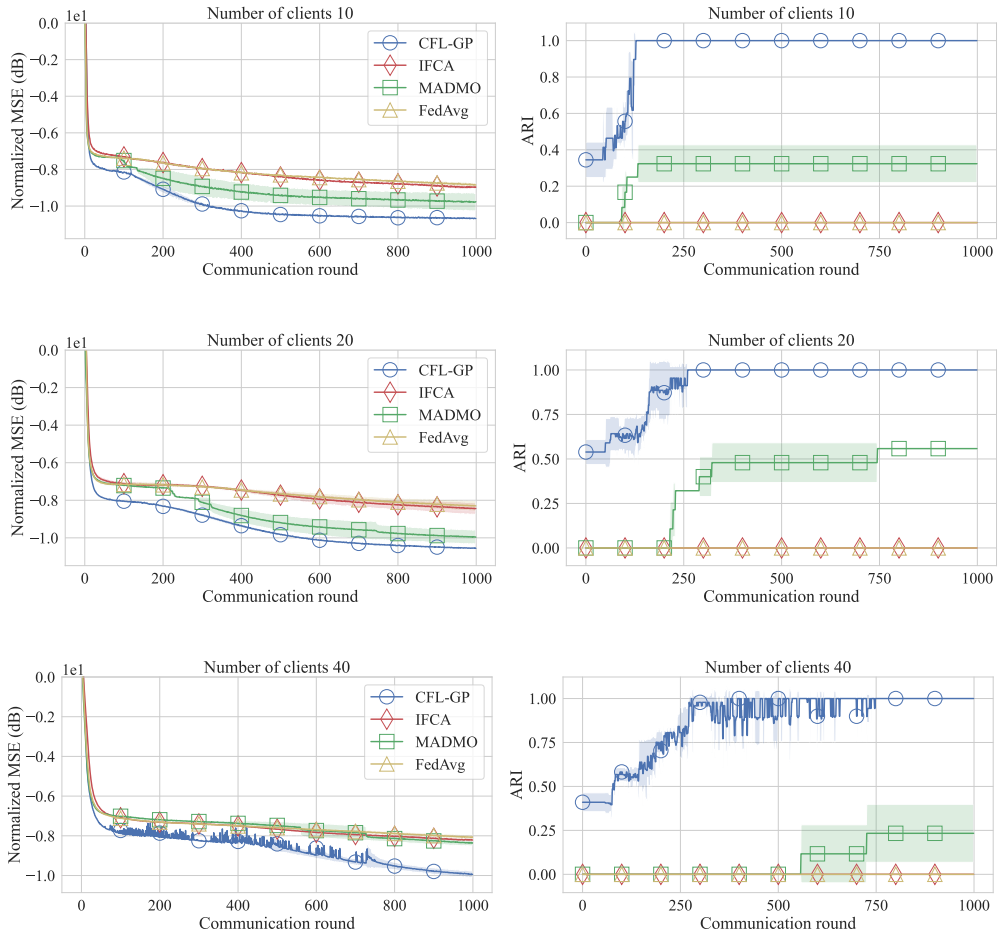

Figure 15. [Deep autoencoder, CD2] Normalized MSE (NMSE) and ARI achieved by each CFL algorithm according to communication round. CFL-GP achieves optimal ARI much faster than the baselines. Despite the highly noisy gradients in this experiment, CFL-GP accumulates gradient information and eventually achieved a perfect ARI of 1.0. In contrast, other algorithms consistently show ARI values below 0.6, resulting in greater distortion (NMSE) compared to the achieved NMSE by CFL-GP.

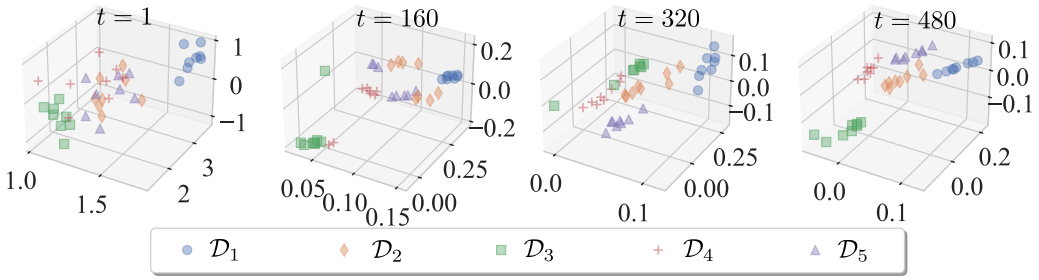

Figure 16. The variations of the client's reduced feature vectors with respect to t (CD2, C=40) are depicted. Each data point represents the reduced feature vector of a client, $\boldsymbol{g}'_c$. It is observed that as t increases, the set of feature vectors from clients corresponding to each distribution identity exhibits a gradual increase in distances between them. As a result, CFL-GP achieves an ARI of 1.0 at $t = 480$.

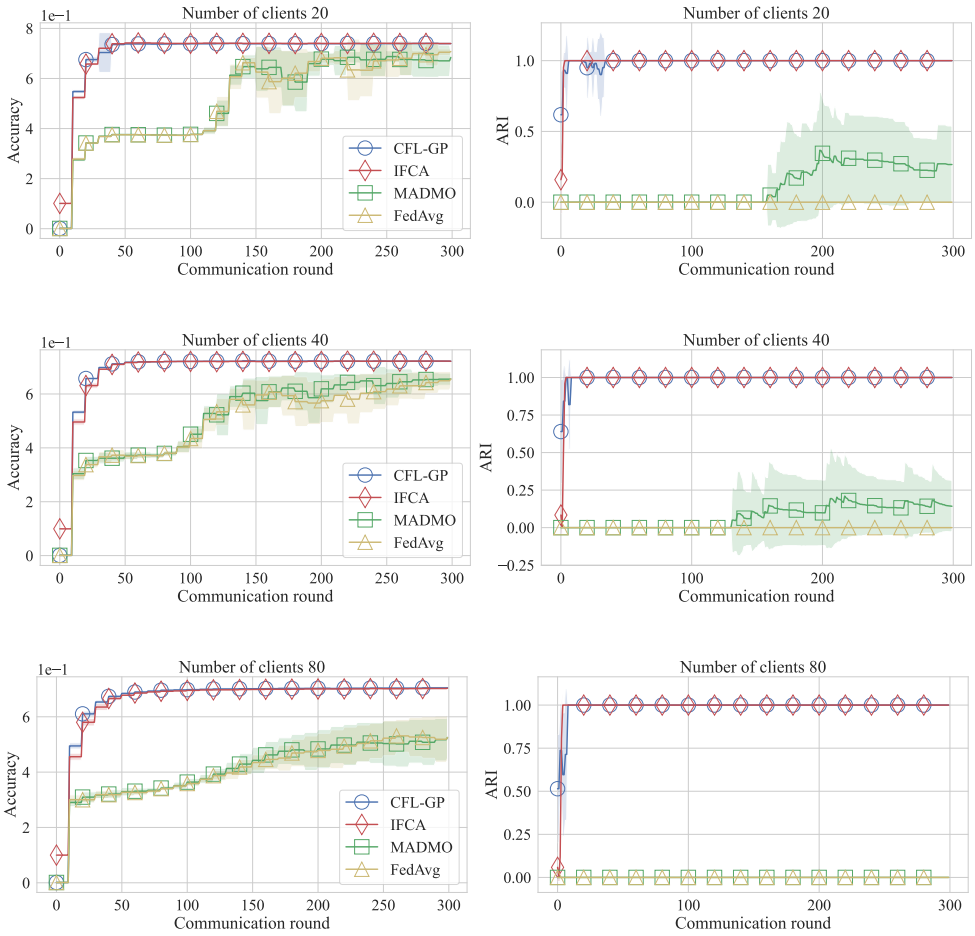

Figure 17. [ResNet18, CIFAR-10] Classification accuracy and ARI over communication rounds. CFL-GP consistently achieves optimal clustering within 50 rounds in all experiments. Notably, CFL-GP achieves a classification accuracy of 60% in less than 30 rounds, while MADMO requires over 140 rounds.

Accuracy, clustering performance, and convergence. In this simulation environment, CFL-GP and IFCA consistently achieve an ARI of 1.0, regardless of the number of clients. This indicates that the loss-based clustering criteria of IFCA, which clusters clients by assigning them to the best-performing model, is effective in this task. MADMO exhibits a slight increase in ARI after 100 rounds but falls short of achieving optimal clustering. When the number of clients is 20, CFL-GP achieves a classification accuracy of 70 or higher in less than 50 iterations. In contrast, MADMO requires more than 250 iterations to reach similar values. These observations also hold true when the number of clients is 40 or 80.

The fast and stable convergence of CFL-GP is primarily due to its ability to achieve optimal clustering quickly. Specifically, CFL-GP attains a maximum ARI of 1.0 within 50 rounds, whereas MADMO requires approximately 200 rounds to reach its peak ARI. By achieving optimal clustering early and updating models for each client cluster, CFL-GP minimizes the noise introduced by gradients from incorrectly clustered clients. This results in more effective model updates, with reduced noise compared to CFL algorithms that struggle with clustering accuracy or FedAvg, which employs a single global model.

F.8. Experiments on EMNIST Dataset

In this subsection, we provide a detailed description of the experiment setup corresponding to **[Exp 3: CNN, EMNIST]** in Table 1 and offer additional analysis corresponding to the results.

We adopt a CFL problem setup in (Sattler, 2020), utilizing the EMNIST dataset (Cohen et al., 2017). The EMNIST dataset

consists of handwritten digits and alphabets, resulting in a total of 62 classes. We randomly sample 200,000 data points for our experiments. The goal of this task is to divide clients with different local datasets based on label heterogeneity and rotation transformations into two distinct groups and train models for each group. The label distributions of the clients vary according to a Dirichlet distribution with a concentration parameter of 1.0. However, the main differentiation for clustering is the rotation transformation. Half of the C clients have datasets subjected to a rotation transformation of 0 degrees, while the remaining clients have datasets subjected to a rotation transformation of 180 degrees. We utilize the convolutional neural network architecture from (Sattler, 2020) and optimize using the cross-entropy loss.

Configuration. We set the number of communication rounds T to 100 and the batch size b to 200. To evaluate performance, we vary the number of clients C to 10, 80, and 160. All algorithms use the model averaging protocol. Local updates are performed with a learning rate of 0.1. The remaining configuration settings follow those specified in Section F.5.

F.8.1. Additional Simulation Results

In this subsection, we investigate the classification performance, and ARI changes with respect to the communication rounds. The first, second, and third rows in Figure 18 correspond to the performance for different numbers of clients: 10, 80, and 160, respectively. The left column shows the classification accuracy over the communication rounds, while the right column depicts the ARI changes over the communication rounds.

From Figure 18, we observe that CFL-GP achieves an ARI of 1.0 in all cases. Particularly, when the number of clients is small, CFL-GP achieves 1.0 ARI within a single clustering period. In other cases, CFL-GP achieves 1.0 ARI within approximately 10 rounds. As a result, CFL-GP also achieves the highest accuracy at $T = 100$. IFCA successfully achieves higher ARI as the number of clients increases. However, when a small number of clients is assumed, IFCA exhibits ARI below 0.75. MADMO consistently shows ARI below 0.75 in all experiments.

F.9. Summary of experiment results

Our extensive experimental investigations aimed to thoroughly assess the performance of CFL-GP with various CFL algorithms, focusing on aspects such as robustness, task performance, convergence speed, and clustering efficacy.

In Appendix D, we presented a detailed analysis of the clustering convergence speed. Additionally, in Appendix E, we demonstrated the robustness of CFL-GP's clustering performance based on compressed or selected gradients. Comparative analyses with the PFL algorithm, FedEM, and PACFL were presented in Appendices F.2 and F.3, respectively. To thoroughly evaluate the algorithms' robustness, we conducted experiments using synthetic datasets, examining their sensitivity to factors such as cluster gaps, batch sizes, and the number of clients. This analysis can be found in Appendix F.4 and Section 5.1. Furthermore, we investigated the clustering capability and convergence of the algorithms on widely used CFL benchmarks, including the MNIST and EMNIST datasets. Our examination encompassed variations in the number of clients, batch sizes, and mixture distribution setups, which can be found in Section 5 and Appendix F.5, F.8. We also explored the applicability of CFL algorithms in scenarios where weight sharing was required for a portion of the parameterized models, as detailed in Appendix F.7. Moreover, we observed the performance of CFL algorithms on industry-standard channel model datasets with deep autoencoders in Appendix F.6.

Throughout our experiments, CFL-GP consistently exhibited optimal clustering performance, surpassing other algorithms in terms of task performance. A particularly significant finding was that CFL-GP is capable of achieving optimal clustering even with just a single round of spectral clustering in many experimental environments. This empirical evidence strongly supports the notion that the gradients obtained from clients contain highly informative knowledge for determining cluster identities. Moreover, our experiments demonstrated that CFL-GP maintains its optimality in some scenarios even when gradients are compressed up to a compression ratio of 10^4, providing further evidence of the rich information it captures.

Our extensive experiments highlight CFL-GP as an effective and practical approach for addressing real-world scenarios requiring optimal clustering, particularly in scenarios that often involve a limited number of clients or data quantities and heterogeneous distributions that are inherently difficult to distinguish.

G. Dealing with Unknown Number of Clusters

Estimating the number of clusters in a dataset without direct access to raw data instances is an open problem that has not been extensively investigated. However, our research has yielded an interesting finding: using the gradient profile matrix of

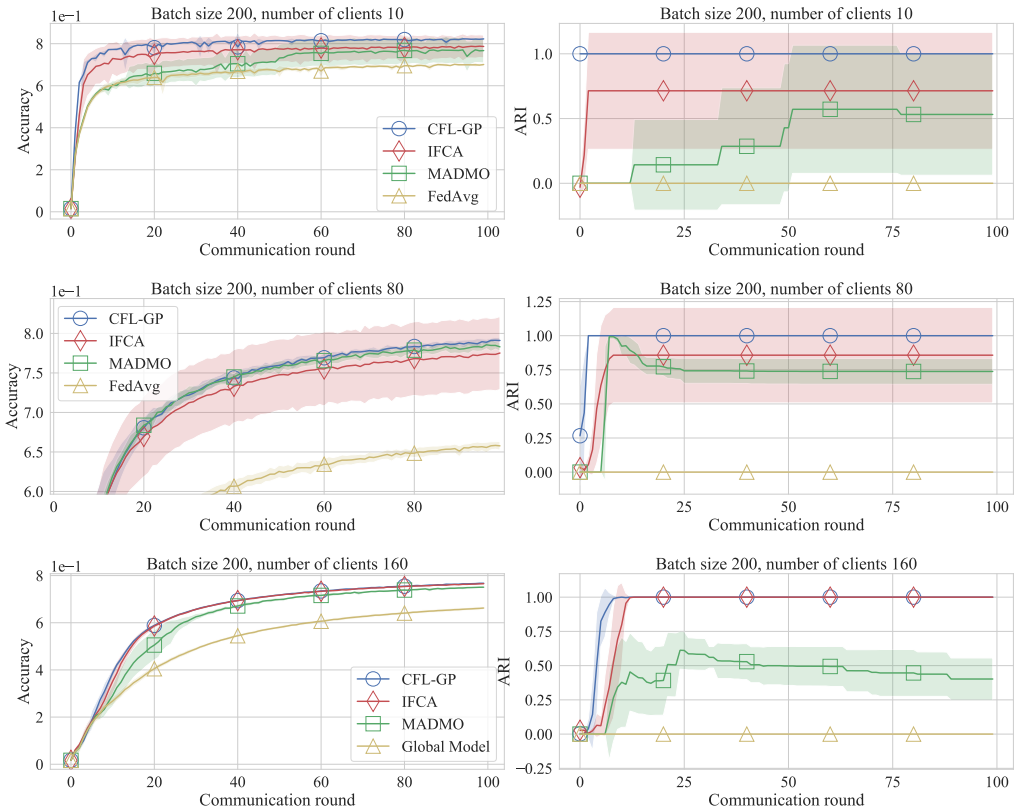

Figure 18. [CNN, EMNIST] Classification accuracy and ARI over communication rounds. CFL-GP consistently achieves optimal clustering within 10 rounds across all experiments, resulting in the highest accuracy performance. However, the other CFL algorithms do not attain optimal clustering within 100 rounds when the number of clients is 10 and 80. When $C = 160$, both CFL-GP and IFCA achieve the optimal clustering, where CFL-GP shows faster convergence toward the optimal clustering compared to IFCA.

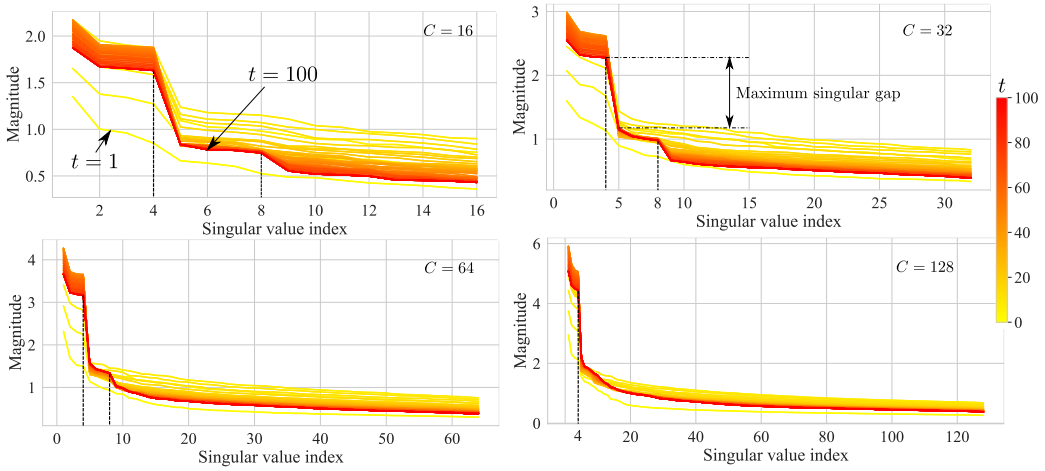

Figure 19. The line plots depict the singular values of the gradient profile matrix at different time points. The intensity of the color, with a deeper shade of red, indicates the results obtained at higher values of t. From left to right and top to bottom, the plots represent the cases where the number of clients is 16, 32, 64, and 128, respectively. In all scenarios, it is evident that the largest gap between consecutive singular values consistently appears between the fourth and fifth singular values.

CFL-GP, we could infer a reasonable number of clusters based on the gradient information in some scenarios.

In this section, we propose a method where we select the s largest singular values from the gradient profile matrix, reduce the client feature to s dimension, and utilize the reduced features to determine the appropriate number of clusters effectively. Note that this approach differs from CFL-GP's dimension reduction of each client's feature vector to a vector of size K, where K is a given number. Instead, it reduces each client's feature to s dimensions, leveraging the leading singular values (or largest spectral gap).

Proposed method. We outline the following steps to deduce the number of clusters: (1) Initially, we arbitrarily select K models and run CFL-GP for a sufficient number of iterations. With each iteration, the gradient profile matrix accumulates more informative data. (2) After a certain number of iterations, we compute the singular values of the gradient profile matrix. By observing the singular values in descending order, we pick the dominant s leading singular vectors. One possible approach is to select the value of s corresponding to the largest singular values beyond which we have a significant spectral gap. (3) Subsequently, we compress the gradient profile matrix $\mathbf{G}$ by employing the s leading singular vectors of $\mathbf{G}$, resulting in dimensionality reduction. Consequently, each client feature vector is reduced to a s-dimensional vector. We can then apply conventional techniques such as the Elbow method or Gap statistic methods on the reduced features to determine the optimal number of clusters.

If the largest s singular values significantly surpass the remaining singular values, it signifies that the gradient directions manifested by C clients within a specific model set can be effectively represented using s dominant directions.

Experiment setup. We used the experiment setup in Appendix F.5, to validate the aforementioned approach to estimating the number of clusters. In the experiment, the MNIST dataset is divided into 8 subsets, each subjected to a specific rotational transformation (0, 15, 90, 105, 180, 185, 270, and 285 degrees), resulting in 8 distinct distribution identities. These subsets are evenly distributed among C clients. We may anticipate two possible results for this setup: (1) we should observe a clear distinction among the eight distribution identities. (2) Alternatively, we can expect to see a clustering that groups together clients with similar rotation transformations (0, 15), (90,105), (180,185), and (270, 285).

Selecting s largest singular values. To empirically determine the number of clusters, we compute the full singular value decomposition (SVD) of the gradient profile matrix of CFL-GP (although the original CFL-GP does not necessarily require a full SVD computation). The resulting singular values are then presented in descending order in Figure 19.

Each subplot in Figure 19 displays the singular values of the gradient profile matrix in descending order, for C=16, 32, 64, and 128 (left to right, top to bottom). The color intensity of the line plots, with a deeper shade of red, indicates results obtained at a greater number of iterations t. By considering the maximum singular gap as the criterion for determining the

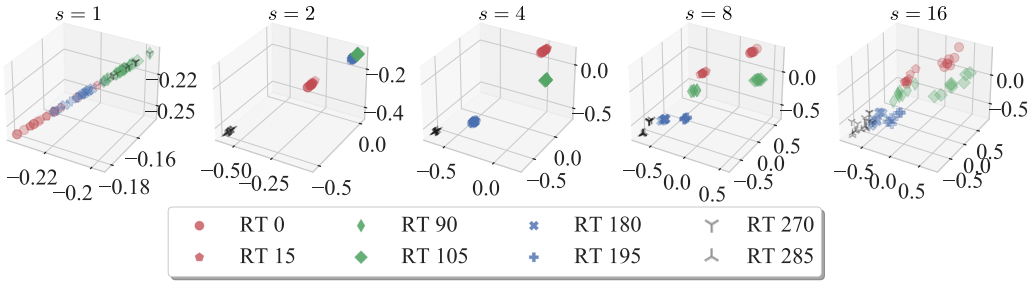

Figure 20. The five subplots display the data points of each client's reduced feature vector $\boldsymbol{g}'_{c,s}$, by s leading left singular vectors. From left to right, the subplots represent the cases where s is 1, 2, 4, 8, and 16, respectively. The distribution identities, based on Rotation Transformations (RTs), are distinguished by different markers. Although the client features are in s dimensions as $\boldsymbol{g}'_{c,s} \in \mathbb{R}^s$, for visualization purposes, they have been randomly projected onto a 3-dimensional space.

number of leading singular vectors, we easily observe that all four subplots exhibit $s = 4$. In other words, the gap between the fourth and fifth singular values is larger than the gaps between other consecutive singular values. If we aim to preserve more information, the observation suggests $s = 8$ as a preferable choice since the next significant gap occurs between the eighth and ninth singular values.

It is important to note that the number of dominant singular values indicates the number of dominant basis vectors to effectively represent the gradient information of clients but does not directly determine the number of clusters. By using the obtained values, $s = 4$ or $s = 8$, we will further reduce the dimensionality of the gradient profile matrix to 4 or 8 dimensions. For comparative analysis, we also arbitrarily select different values of s (s=1, 2, 16) to compare the results.

Let us denote the gradient profile matrix we are dealing with as $\mathbf{G} = \mathbf{G}^{(t)} = (\boldsymbol{g}_1^{(t)}, \cdots, \boldsymbol{g}_C^{(t)})$. We can reduce each client feature to s dimensions using the left singular vector matrix of $\mathbf{G}$. Consider a SVD for the matrix $\mathbf{G}$ as follows.

$$\mathbf{G} = \sum_{i=1}^{C} \hat{\sigma}_i \hat{\boldsymbol{u}}_i \hat{\boldsymbol{v}}_i^\top \tag{81}$$

where $\hat{\sigma}_i$ represents the i-th largest singular value of $\mathbf{G}$, $\hat{\boldsymbol{u}}_i$ and $\hat{\boldsymbol{v}}_i$ denote the left and right singular vectors corresponding to $\hat{\sigma}_i$, respectively. We then define $\hat{\boldsymbol{U}}_s = (\hat{\boldsymbol{u}}_1, \cdots, \hat{\boldsymbol{u}}_s)$ for $s \leq C$. By using this matrix, we reduce the dimensionality of the client feature vectors as $\boldsymbol{g}'_{c,s} = \hat{\boldsymbol{U}}_s^\top \boldsymbol{g}_c \in \mathbb{R}^s$.

Results and Analysis. Figure 20 represents the dimension-reduced client feature vectors $\boldsymbol{g}'_{c,s}$ based on different values of s when $t = 20$ and $C = 64$. The five subplots show the results for s=1, 2, 4, 8, and 16, respectively, from left to right. The data points in each subplot represent the reduced client feature vectors, denoted as $\boldsymbol{g}'_{c,s}$. The s-dimensional client feature vectors are randomly projected onto a 3-dimensional space for visualization. Different markers are applied to indicate the distribution identity of each client.

From Figure 20, we can infer reasonable cluster results based on $s = 4, 8$ obtained from the previous Figure 19. When setting $s = 8$, we can observe that different sets of clients corresponding to the eight distribution identities are accurately distinguished (subplot $s = 8$). This is because the distances between reduced feature vectors of clients with the same distribution identity are smaller compared to the distances between feature vectors of clients with different distribution identities. In other words, when compressing the feature vectors using only the dominant eight basis vectors of the column space of $\mathbf{G}$ corresponding to the leading eight singular values, all eight distinct distribution identities can be fully recovered, as depicted in the fourth subplot.

If we consider only the four dominant directions from the gradient profile matrix, we obtain the third subplot ($s = 4$). This result provides a clustering outcome where clients with the most similar rotation transformations are grouped together.

However, if we set $s = 1$ or $s = 2$ (i.e., if we lose some of the dominant directions in the column space of the gradient profile matrix), as seen in the first two subplots, we may fail to accurately estimate the cluster identities. Indeed setting $s = 1$, we may not be able to cluster the data points belonging to RT 270 and 285. Similarly, when setting $s = 2$, the clients corresponding to RT 90 to 195 are grouped together, leading to an undesirable clustering result.

In the final scenario, when we choose $s = 16$ to incorporate a larger number of basis vectors from the gradient profile matrix and examine the scattered pattern of the reduced feature vectors, it becomes challenging to observe a clear distinction between the distribution identities. The high dimensionality of the feature vectors may lead to overlapping clusters, making it difficult to determine a definite number of clusters based on this analysis.

According to this empirical analysis, if a learning system engineer wishes to determine the number of models based on estimated values rather than a pre-determined value of K, it is expected that by running CFL-GP for a certain duration with an arbitrary K and subsequently conducting the singular value analysis described earlier, a reasonable estimate of the value D can be obtained.