One-Shot Federated Learning for Model Clustering and Learning in Heterogeneous Environments

Aleksandar Armacki  
Carnegie Mellon University, Pittsburgh, PA  
aarmacki@andrew.cmu.edu

Dragana Bajovic  
Faculty of Technical Sciences, University of Novi Sad, Novi Sad  
dbajovic@uns.ac.rs

Dusan Jakovetic  
Faculty of Sciences, University of Novi Sad, Novi Sad  
dusan.jakovetic@dmi.uns.ac.rs

Soummya Kar  
Carnegie Mellon University, Pittsburgh, PA  
soummyak@andrew.cmu.edu

Abstract

The paper presents a communication efficient approach for federated learning in heterogeneous environments in which users obtain data from one of $K$ different data distributions. In the proposed setup, the number $K$ of data distributions and their underlying statistical properties as well as the user cluster structure (i.e., the grouping of users based on the data distributions they sample) are apriori unknown. A one-shot decentralized learning approach, based on a single round of communication between the users and the server, is proposed with the objective of learning the true model at each user. The proposed one-shot approach, based on local computations at the users and a convex clustering based aggregation step at the server is shown to provide strong learning guarantees in such heterogeneous environments. In particular, it is shown that for strongly convex learning setups, as long as the number of data points per user is above a threshold, the proposed approach achieves order-optimal mean-squared error (MSE) rates in terms of the sample size with respect to a hypothetical oracle that has access to all data points at all users and perfect information about the
number of different distributions and the user cluster structure, i.e., assignment of distributions to users. An explicit characterization of the threshold is provided in terms of the problem parameters. Numerical experiments illustrate the findings and corroborate the performance of the proposed method.

1 Introduction

Federated learning (FL) is a paradigm where many users collaborate, with the aim of jointly training a model [1]. While such an approach helps alleviate the storage and computation burden for any single user, it imposes significant communication costs on the system as a whole, e.g., [2]. To tackle this issue, different approaches have been proposed, such as quantization, e.g., [3], [4], [5], local methods, e.g., [6], [7], [8] and one-shot methods, e.g., [9], [10], [11], [12], [13], to name a few. Another issue associated with training a global model comes from system heterogeneity. Users that participate in FL often contain datasets generated by different distributions, making the system as a whole highly heterogeneous. A global model can therefore be very bad for an individual user [14], [15].

Many different approaches that address the shortcomings of the global model have been proposed. One such approach is personalized federated learning (PFL). The goal of PFL is to learn models that fit each individual user, while utilizing the federation to produce models that generalize better. Many approaches to personalization have been proposed, such as multi-task learning [16], [17], [18], meta-learning [19], [20], fine-tuning [21], [22].

Another closely related approach is clustered federated learning (CFL). The underlying assumption in clustered federated learning is the presence of $K$ different data distributions $\mathcal{D}_k$, $k \in [K]$, with each user sampling their data from only one of $K$ distributions. This leads to a natural clustering of users, i.e., we can define clusters $\{C_k\}_{k \in [K]}$, given by $C_k = \{i \in [m] : \text{user } i \text{'s data follows distribution } \mathcal{D}_k\}$ and $\cup_{k \in [K]} C_k = [m]$ and $C_k \cap C_j = \emptyset$, $\forall k \neq j$. In such a scenario the goal is to learn $K$ models associated with the underlying clusters, so that users belonging to the same clusters have the same models. This is somewhat different from the classical PFL approaches,

This work was supported in part by the Office of Naval Research under Grant No N00014-21-1-2547 and the National Science Foundation under Grant No CNS-1837607. This work is also supported by the European Union’s Horizon 2020 Research and Innovation program under grant agreement No 957337. The paper reflects only the view of the authors and the Commission is not responsible for any use that may be made of the information it contains.
where the goal is to produce $m$ models, one for each user. Allowing for $1 \leq K \leq m$, clustered federated learning can again be seen as an intermediary between the global and local learning, with $K = 1$ resulting in a global model, while $K = m$ resulting in purely local models. There are many works assuming a clustered structure among users, such as [23], [24], [25], [26], [27].

While existing works in CFL focus on dealing with heterogeneity and personalization aspects, none of them focus on communication efficiency. The aim of this paper is to provide a method for CFL that maintains the benefits of standard clustering-based approaches, while simultaneously achieving communication efficiency. This is achieved by developing a one-shot federated learning method, that requires only one round of communication.

**Literature review.** We next review the related literature, in particular, one-shot methods in FL and methods for CFL.

**One-shot methods** in the context of FL have been studied in [9], [10], [11], [12], [13]. [9] study one-shot averaging methods. Each user trains a model on the local data and the server produces the final model by averaging all the users’ models. The authors show that, under certain assumptions, the methods can achieve the same MSE guarantees as centralized learning, i.e., order-optimal rates in terms of sample size, provided that the number of samples available to each user is higher than a threshold. [10] proposes to train $K \leq m$ ensemble based methods for supervised and semi-supervised problems. [11] propose a one-shot distillation method, wherein the users send a distilled version of their local dataset to the server, which then performs the global model training. [12] study one-shot methods in federated settings under constraints on the communication budget. The proposed method is, under certain regimes, order-optimal up to logarithmic factors, while simultaneously relaxing the higher-order smoothness assumptions made in [9]. [13] introduces a one-shot FL method in heterogeneous settings, designed specifically for data clustering. The methods in [9] and [12] provide strong theoretical guarantees, however, they assume that the data across all users follows a single distribution $\mathcal{D}$. Therefore, they focus on training a single global model to be used for all users. In modern FL systems the data across different users typically comes from different distributions, hence violating the identically distributed data assumptions made in prior works. Moreover, the user heterogeneity stemming from this phenomena is known to hamper the global model [14]. The methods [10] and [11] consider heterogeneous settings, but provide no theoretical analysis. **To the best of our knowledge,**

---

1. The method [13] provides a theoretical analysis under heterogeneous settings. However, the method is not a general learning method, but a method designed for clustering.
no theoretical results for one-shot methods are established under the presence of heterogeneity, i.e., multiple data distributions in the system.

CFL has been studied in [23], [24], [25], [26], [27]. [23] and [24] propose mutually similar methods, that iteratively estimate cluster membership and perform model updates. The authors in [24] show an exponential convergence rate up to an error floor that is order-optimal up to a logarithmic factor and dependence on the smallest cluster size. [25] propose a robust algorithm for CFL, under the presence of adversarial users. If there are no adversarial users (the setting that we consider in this paper), the method is order-optimal up to a logarithmic factor. [26] propose a method for CFL that can be applied to any standard FL method, as a fine-tuning step. The method is based on successive bi-partitioning of the current set of users, based on cosine similarity and does not require prior knowledge of the number of clusters K. However, when a bi-partitioning is performed, each partition needs to do a full FL training on the newly formed partition/federation. Finally, [27] propose a method that aims to simultaneously infer the clustering of users and train models. The proposed method does not require knowledge of K and establishes explicit conditions under which the true clustering can be recovered. All of the methods require potentially many rounds of communication and model training. Among those, the best is achieved by [24], requiring \( O\left(\frac{\kappa}{p} \log \left(\frac{2D}{\varepsilon}\right)\right) \) communication rounds, where \( \kappa \) is the condition number, \( D \) is a parameter that characterizes the similarity of different distributions (to be defined in Assumption 1 ahead), \( p \) represents the fraction of users belonging to the smallest cluster, while \( \varepsilon > 0 \) is the final accuracy threshold achieved by the method. The methods [23], [24] and [25] require prior knowledge of the number of clusters K. While the methods in [26] and [27] do not require knowledge of K, they require many rounds of communication.

Contributions. In this paper we propose a one-shot method for CFL, that is able to deal with system heterogeneity. We study the statistical guarantees of the method, in terms of convergence rates with respect to the number of samples, as in [9], [12], [24] and [25]. Our contributions can be summarized as follows:

1. We propose a one-shot method for CFL under the presence of multiple data distributions (i.e., system heterogeneity). The method is communication efficient, by only requiring one round of communication. Moreover, the proposed method does not require knowledge of the true number of clusters K.

2. We show that, for strongly convex costs, the method achieves the order-
optimal MSE guarantees in terms of sample size, i.e., it matches the order-optimal MSE guarantees of centralized learning, provided that each user has a number of data points above a threshold, establishing regimes in which communication beyond the first round is not necessary for achieving order-optimality.

3. We show that, compared to existing methods, our algorithm reduces communication cost by a factor of \( O\left(\frac{n}{p} \log\left(\frac{2D}{\varepsilon}\right)\right) \), while improving the rates by a factor logarithmic in the number of samples and users.

4. We explicitly derive the expression for the threshold on the number of data points for the users to achieve the order-optimal MSE rate and show how it depends on various system parameters, e.g., the size of clusters, difficulty of the clustering problem and other problem related parameters.

5. We verify our theoretical findings via numerical experiments, showing the proposed method achieves the order-optimal MSE rate and matches the performance of oracle methods that know the true cluster membership.

**Paper organization.** The rest of the paper is organized as follows. Section 2 introduces the relevant background and formally states the problem. Section 3 describes the proposed method. Section 4 presents the main results of the paper. Section 5 presents numerical results. Finally, Section 6 concludes the paper. The Appendix contains some additional theoretical results. The reminder of the section introduces the notation used throughout the paper.

**Notation.** The set of real numbers is denoted by \( \mathbb{R} \), while \( \mathbb{R}^d \) denotes the corresponding \( d \)-dimensional vector space. \( \mathbb{N} \) denotes the set of positive integers. \( \langle \cdot, \cdot \rangle \) denotes the Euclidean inner product and \( \| \cdot \| \) denotes the induced norm. In a slight abuse of notation, we will also use \( \| \cdot \| \) to denote the matrix norm induced by the Euclidean norm. \([m]\) denotes the set of positive integers up to and including \( m \in \mathbb{N} \), i.e., \([m] = \{1, 2, \ldots, m\}\). We use \( \{S_k\}_{k \in [K]} \) to denote a collection of sets indexed by \([K]\), i.e., \( \{S_k\}_{k \in [K]} = \{S_k : k \in [K]\}\). For a collection of sets \( \{S_k\}_{k \in [K]} \), we use \( S_{(k)} \) to denote the \( k \)-th largest set. We use \( \text{int} A \) to denote the interior of set \( A \). The notation \( O(\cdot) \), \( \Omega(\cdot) \) refers to the standard “big O” and “big Omega” notation, respectively, i.e., for two non-negative sequences \( \{a_n\}_{n \in \mathbb{N}} \) and \( \{b_n\}_{n \in \mathbb{N}} \), the relation \( a_n = O(b_n) \) (\( a_n = \Omega(b_n) \)) implies the existence of a global constant \( C_1 > 0 \) and \( n_1 \in \mathbb{N} \), such that \( a_n \leq C_1 b_n \) (\( a_n \geq C_1 b_n \)), for all \( n \geq n_1 \).
2 Problem formulation and preliminaries

In this section, we define the problem of interest. We introduce some assumptions and discuss their implications. In Subsections 2.1 and 2.2 we introduce the method from [9] and convex clustering, respectively.

Consider $m$ users, $i = 1, \ldots, m$, that participate in a federated learning system. Each user $i$ contains a local dataset $x_{ij} \in \mathcal{X}$, $j = 1, \ldots, n_i$, distributed according to $\mathcal{D}_i$. Here, $n_i \in \mathbb{N}$ represents the number of samples available to user $i$ and $\mathcal{X} \subset \mathbb{R}^d$ represents the data space. Given a loss function $\ell : \Theta \times \mathcal{X} \mapsto \mathbb{R}$, each user forms the local empirical loss $f_i : \Theta \mapsto \mathbb{R}$, given by

$$f_i(\theta) = \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(\theta; x_{ij}).$$

(1)

The local empirical loss represents an estimate of the population loss $F_i : \Theta \mapsto \mathbb{R}$, given by

$$F_i(\theta) = \mathbb{E}_{X_i \sim \mathcal{D}_i}[\ell(\theta; X_i)],$$

(2)

where $X_i$ is the data generating random variable distributed according to $\mathcal{D}_i$. The goal of each user is to train a model that performs well with respect to the population loss (2), using only the local empirical loss (1) and leveraging the empirical losses of similar users. To make this notion more precise and formally introduce the problem we want to solve, we now state the assumption used throughout the paper.

**Assumption 1.** There exist $K$ different data distributions in the system, with $1 < K < m$, such that each user samples independent, identically distributed (IID) data from only one of the distributions, i.e., for each $i \in [m]$, we have $\mathcal{D}_i = \mathcal{D}_k$, for some $k \in [K]$. The data sampled across users is also independent. Denote by $D$ the minimal distance between population optimal models of each cluster, $\theta^*_k := \arg\min_{\theta_k \in \Theta} F_k(\theta_k), k \in [K]$, i.e., $D = \min_{k \neq l} \|\theta^*_k - \theta^*_l\| > 0$. We then have $D > 0$.

Assumption [1] provides a natural partitioning of the set of all users $[m]$, given by $\mathcal{C} = \{C_k\}_{k \in [K]}$, where $C_k$’s are mutually disjoint and their union is the whole set $[m]$.

**Remark 1.** Assumption [1] can be interpreted as a measure of distance between different distributions. Intuitively, it states that, for $D$ large, the optimal model corresponding to one of the $K$ different populations will not be a good model for any other population.
Under Assumption 1, the models offering optimal statistical guarantees can be trained by solving

\[
\arg\min_{\theta_1, \ldots, \theta_K} \sum_{k=1}^K \frac{|C_k|}{m} g_k(\theta_k),
\]

where \( g_k : \Theta \mapsto \mathbb{R} \) is the cluster-wise loss, \( g_k(\theta) = \frac{1}{|C_k|} \sum_{i \in C_k} f_i(\theta) \). To see why such an approach is optimal recall that, for a user \( i \in C_k \), the optimal population model is given by \( \theta_k^* = \arg\min_{\theta_k \in \Theta} F_k(\theta_k) \). If we denote the empirical risk minimizers (ERMs) as \( \hat{\theta}_i = \arg\min_{\theta_i \in \Theta} f_i(\theta_i) \), applying results from, e.g., [28], gives \( \mathbb{E} \| \theta_k^* - \hat{\theta}_i \|^2 = O \left( \frac{1}{n_i} \right) \). Let \( \hat{\theta}_k \) be minimizer of the empirical cluster-wise cost, i.e., \( \hat{\theta}_k = \arg\min_{\theta_k \in \Theta} g_k(\theta_k) \). We then have \( \mathbb{E} \| \theta_k^* - \hat{\theta}_k \|^2 = O \left( \frac{1}{n_k} \right) \), where \( n_k = \sum_{i \in C_k} n_i \) is the total number of available IID samples from \( D_k \), showing the benefits of clustered learning. Actually, the \( O \left( \frac{1}{n_k} \right) \) MSE decay is the best achievable in our setting. See, e.g., the Hajek-Le Cam minimax theorem [29, Theorem 8.11] for a formal account of the statement; see also [9]. On the other hand, by Assumption 1 and cluster design, the benefits of further merging (and/or modifying) the clusters, in terms of sample size, can potentially be significantly outweighted by the distribution skew between two different clusters.

We now state the rest of the assumptions used throughout the paper.

**Assumption 2.** The parameter space \( \Theta \subset \mathbb{R}^d \) is a compact convex set, with \( \theta_k^* \in \text{int} \Theta \), for all \( k \in [K] \). Let \( R > 0 \) be such that \( \| \theta \| \leq R \), for all \( \theta \in \Theta \).

**Remark 2.** Assumption 2 is a standard assumption on the parameter space in the literature, e.g., [9], [28], [30].

**Assumption 3.** For any fixed \( x \in \mathcal{X} \), the loss function \( \ell(\cdot; x) : \Theta \mapsto \mathbb{R} \) is: 1) nonnegative, i.e., for any \( \theta \in \Theta \), \( \ell(\theta; x) \geq 0 \); 2) convex, i.e., for any \( \theta, \theta' \in \Theta \), we have \( \ell(\theta'; x) \geq \ell(\theta; x) + \langle \nabla \ell(\theta; x), \theta' - \theta \rangle \); 3) \( L \)-smooth, i.e., for a fixed constant \( L > 0 \) and any \( \theta, \theta' \in \Theta \), we have \( \ell(\theta'; x) \leq \ell(\theta; x) + \langle \nabla \ell(\theta; x), \theta' - \theta \rangle + \frac{L}{2} \| \theta' - \theta \|^2 \).

**Remark 3.** Note that the constant \( L \) for the smoothness condition is independent of the choice of \( x \), i.e., \( L \) is a global constant that holds for any choice of \( x \in \mathcal{X} \).

**Remark 4.** Note that Assumption 3 implies non-negativity, convexity and \( L \)-smoothness of all of \( F_k, g_k, f_i, k \in [K], i \in [m] \). Moreover, under convexity
of \( \ell \), the smoothness condition is equivalent to Lipschitz continuous gradient of \( \ell \), i.e., for any fixed \( x \in \mathcal{X} \) and any \( \theta, \theta' \in \Theta \), we have \( \|\nabla \ell(\theta; x) - \nabla \ell(\theta'; x)\| \leq L\|\theta - \theta'\| \).

**Assumption 4.** For each \( k \in [K] \), the population loss \( F_k(\theta) = \mathbb{E}_{X_k \sim D_k}[\ell(\theta; X_k)] \) is strongly convex, i.e., there exists a constant \( \mu_{F_k} > 0 \), such that, for all \( \theta, \theta' \in \Theta \), we have \( F_k(\theta') \geq F_k(\theta) + \langle \nabla F_k(\theta), \theta' - \theta \rangle + \frac{\mu_{F_k}}{2}\|\theta - \theta'\|^2 \). Denote by \( \mu_F = \min_{k \in [K]} \mu_{F_k} \).

**Assumption 5.** For each \( k \in [K] \), there exists a neighborhood \( U_k = \{ \theta \in \Theta : \|\theta - \theta_k^*\| \leq \rho_k \} \), where \( \rho_k > 0 \) and constants \( S_k, H_k > 0 \) such that, for all \( x \in \mathcal{X} \) and any \( \theta, \theta' \in U_k \), the loss \( \ell \) satisfies \( \mathbb{E}_{X_k \sim D_k}[\|\nabla \ell(\theta; X_k)\|^8] \leq S_k^8 \) and \( \|\nabla^2 \ell(\theta; x) - \nabla^2 \ell(\theta'; x)\| \leq P_k\|\theta - \theta'\| \).

**Remark 5.** Assumption\(^\text{5}\) requires each population loss \( \ell \) to be well-behaved in a neighborhood of the optimal model. Akin to Assumption 3 in \([9]\), this assumption is required for averaging methods to work. We refer the reader to \([9]\) and references therein, for an elaborate discussion on this requirement.

From Remark \(^\text{4}\) we can see that, for each fixed \( x \in \mathcal{X} \), the gradients of \( \ell \) are continuous. Using the compactness of \( \Theta \), we can conclude that \( \ell \) has bounded gradients over \( \Theta \), for any fixed \( x \in \mathcal{X} \). Denote by \( S(x) \) the global gradient bound at \( x \in \mathcal{X} \), i.e., \( S(x) = \sup_{\theta \in \Theta} \|\nabla \ell(\theta; x)\| \). Next, combining Remark \(^\text{4}\) and the compactness of \( \Theta \), we can see that \( F_k \)'s and \( f_i \)'s have bounded gradients, for all \( k \in [K], i \in \mathcal{I} \). Denote the bounds by \( G_{F_k} = \max_{\theta \in \Theta} \|\nabla F_k(\theta)\| \) and \( G_{f_i} = \max_{\theta \in \Theta} \|\nabla f_i(\theta)\| \). Appealing to the mean value theorem, we can see that \( F_k \)'s and \( f_i \)'s are all Lipschitz continuous, with constants \( G_{F_k} \) and \( G_{f_i} \), respectively. From the \( L \)-Lipschitz continuous gradients of \( \ell \) and the continuity of the Hessian of \( \ell \) on \( U_k \), we can conclude that the Hessian of \( \ell \) is uniformly bounded on \( U_k, k \in [K] \), with the bounding constant being equal to \( L \), i.e., \( L = \max_{x \in \mathcal{X}, \theta \in U_k} \|\nabla^2 \ell(\theta; x)\| \), for all \( k \in [K] \).

Assumptions\(^\text{2,5}\) are standard in the literature, e.g., the reader is referred to \([9], [30], [24]\) and the references therein.

Note that the formulation \(^\text{3}\) implicitly assumes the knowledge of the true clustering structure. In reality, the distributions and their associated clustering structures are not known. Moreover, even the exact number of different distributions, \( K \), is typically not available. Therefore, the formulation \(^\text{3}\) is impossible to obtain and solve in practice. In what follows, we propose a method that is able to deal with these issues, by correctly identifying the true clusters and producing models that offer the same order-optimal
MSE guarantees, as the models with knowledge of the true clustering structure, obtained by (3).

2.1 The AVGM method

The authors in [9] study the problem (3), under the assumption that all the distributions are the same, i.e., \( D_k = D \), for all \( k \in [K] \). They propose the following two-step method, called AVGM, that requires only one round of communication: 1) each user \( i \) obtains a local model \( \hat{\theta}_i \), by solving \( \hat{\theta}_i = \arg \min_{\theta \in \Theta} f_i(\theta) \) and sends it to the server; 2) the server receives the local models and produces the final model by averaging, i.e., \( \overline{\theta} = \frac{1}{m} \sum_{i=1}^{m} \hat{\theta}_i \).

Assuming \( n_i = n, i \in [m] \), for some \( n \in \mathbb{N} \), the authors show that, when \( n \geq m \), the method results in the order-optimal MSE, i.e., \( E \| \theta - \theta^* \|^2 = O \left( \frac{1}{mn} \right) \).

Here, \( \theta^* = \arg \min_{\theta \in \Theta} F(\theta) \) is the optimal model for the entire population.

2.2 Convex clustering

Convex clustering is a well-studied approach to clustering, e.g., [31], [32], [33], wherein the clustering problem is formulated as a strongly convex optimization problem with group lasso regularization. As such, the method is guaranteed to have a unique solution and, moreover, does not require knowledge of the true number of clusters \( K \). Formally, for a given dataset \( A = \{a_1, \ldots, a_n\} \), with \( a_i \in \mathbb{R}^d \), the problem of convex clustering is formulated as

\[
\arg \min_{u_1, \ldots, u_n \in \mathbb{R}^d} \frac{1}{2} \sum_{i=1}^{n} \|a_i - u_i\|^2 + \lambda \sum_{i<j} \|u_i - u_j\|,
\]

where \( \lambda > 0 \) is a tunable parameter. Let \( V = \{V_k\}_{k \in [K]} \) be a partition of \( A \), such that \( \bigcup_{k \in [K]} V_k = A \) and \( V_k \cap V_l = \emptyset, k \neq l \). [33] Corollary 7 show that, if \( \lambda \) satisfies

\[
\max_{k \in [K]} \frac{diam(V_k)}{|V_k|} \leq \lambda < \min_{k, l \in [K]} \frac{\|c(V_k) - c(V_l)\|}{2n - |V_k| - |V_l|},
\]

the clustering, is recovered, in the sense that, we have \( u_i^* = u_j^* \), for all \( i, j \in V_k \) and \( u_i^* \neq u_j^* \), for all \( i \in V_k, j \in V_l, k \neq l \). Here, \( \{u_i^*\}_{i=1}^{n} = \{u_i^*(\lambda)\}_{i=1}^{n} \) is the optimal solution produced by [4], \( diam(S) = \max \{\|x - y\| : x, y \in S\} \) is the diameter of a set \( S \subset \mathbb{R}^d \), while \( c(S) = \frac{1}{|S|} \sum_{x \in S} x \) is the centroid of \( S \).

Remark 6. One can also implement a weighted version of convex clustering,
given by

\[
\frac{1}{2} \sum_{i=1}^{n} \|a_i - u_i\|^2 + \lambda \sum_{i<j} w_{ij} \|u_i - u_j\|,
\]

where \( w_{ij} \geq 0 \) are weights assigned to the pair \((i, j)\). The weighted version of convex clustering has been observed to perform well in practice, e.g., [31], [32], [34], and can reduce the complexity of the convex clustering problem, by setting many \( w_{ij} \)'s to zero. However, recovery guarantees of weighted convex clustering require that across cluster weights for different clusters (i.e., such that \( i \in C_k, j \in C_l, k \neq l \)) be non-zero, e.g., [33]. Therefore, it is not obvious which \( w_{ij} \)'s can be set to zero without knowing the true clustering structure beforehand. While we chose to present our algorithm using the uniformly-weighted convex clustering method, i.e., \( w_{ij} = 1 \), for all \( i < j \), for the previously outlined reason, we note that a weighted version of convex clustering with a heuristic \( k \)-Nearest Neighbor weight selection as in, e.g., [33], can be implemented with our algorithm.

3 Algorithm design

In this section, we outline our one-shot algorithm for FL in heterogeneous environments. Subsection 3.1 describes the proposed one-shot method. Subsection 3.2 outlines some considerations when applying the method in practice.

3.1 The proposed method

In order to deal with the presence of multiple data distributions, we propose a method that works as follows:

1. Each user \( i \) obtains a local model \( \hat{\theta}_i \), by solving \( \hat{\theta}_i = \arg\min_{\theta_i \in \Theta} f_i(\theta_i) \) and sends it to the server.

2. The server receives the local models \( \{\hat{\theta}_i\}_{i=1}^{n} \), chooses a value \( \lambda \in \left[ \sqrt{\frac{2M \log n}{n}}, \frac{|C_K|((D-2)\gamma)}{2m-|C_{K-1}|-|C_K|} \right] \) (see Theorem 1 ahead) and runs the convex clustering algorithm [4], with the local models as inputs, resulting in \( K' \) clusters \( C' = \{C'_{k'}\}_{k' \in [K']} \).

3. The server then averages the local models according to the resulting clusters, i.e., for each obtained cluster \( C'_{k'} \), \( k' \in [K'] \), the server performs \( \bar{\theta}_{k'} = \frac{1}{|C'_{k'}|} \sum_{i \in C'_{k'}} \hat{\theta}_i \).
4. The server then sends the models to each user, corresponding to their cluster assignment, i.e., each user $i \in C'_{k'}$ receives the model $\hat{\theta}_{k'}$.

Note that the main difference between the method in [9] and the proposed method is in step 2, where the server performs clustering of the models. This step is necessary, as we aim to identify the true clustering structure, and produce a model that maintains the guarantees of the clustered approach (3). We chose the convex clustering method, e.g., [31], [32], [33], as it does not require knowledge of the exact number of clusters $K$. Note that, if knowledge of the number of clusters was available, a simpler algorithm, like K-means, e.g., [35], [36], or gradient clustering, e.g., [37], can be applied.

### 3.2 Practical considerations

Notice that the upper bound in the algorithm above,

$$\lambda_u = \frac{|C(K)| (D - 2\gamma)}{2m - |C(K)| - |C(K-1)|},$$

requires the knowledge of the true clustering. Moreover, the lower and upper bounds in the recovery condition (5) both depend on the recovered clustering, which in turns depends on the value of $\lambda$, via (4). This shows that (5) (and (6) ahead) can only be verified in "a posteriori" manner, after (4) is solved. Therefore, choosing an appropriate value of $\lambda$ can be difficult in practice. In this subsection we provide an algorithm that includes practical guidelines on choosing an appropriate value of the parameter $\lambda$, elaborating on step 2) from the previous subsection. The algorithm performs the same steps 1, 3, 4 as outlined in the algorithm from Subsection 3.1. The step 2 is modified as follows:

2. The server receives the local models $\{\hat{\theta}_i\}_{i=1}^m$ and chooses a range of strictly increasing values of $\lambda$, $\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$, such that solving the convex clustering problem (4) results in the number of clusters $K_{\lambda}$ satisfying $K_{\lambda_1} = m$ and $K_{\lambda_N} = 1$. The server runs the convex clustering algorithm for each value of $\lambda_i$ and verifies the following:

(a) If the condition (5) is verified for some values of $\lambda_i$, the server takes a value of $\lambda_i$ (and the associated clustering) such that the same number of clusters $K_{\lambda_i}$ is recovered for the largest number of $\lambda_i$'s, among all of $\lambda_i$'s, $i = 1, \ldots, N$.

From the formulation of convex clustering (4), it is obvious that, for $\lambda$ sufficiently small, the optimal solution is going to be $u'_i = a_i$, $i \in [m]$, i.e., $K_{\lambda} = m$. On the other hand, the authors in [38] show that, for $\lambda$ sufficiently large, we have $K_{\lambda} = 1$. Hence, the choices of $\lambda$ guaranteeing $K_{\lambda} = m$ and $K_{\lambda} = 1$ always exist.
(b) If the condition (5) is not verified for any value of $\lambda_i$, the server takes a value $\lambda_i$ (and the associated clustering) such that the same number of clusters $K_{\lambda_i}$ is recovered for the largest number of $\lambda_i$’s, among all of $\lambda_i$’s, $i = 1, \ldots, N$.

The procedure in step 2 is designed in the spirit of "clusterpath", e.g., [32]. The intuition behind the procedure is to choose the clustering that is the likeliest to be true. If (5) is verified for some $\lambda$’s, the likelihood of the recovered clustering matching the true one is increasing. Note that in general, the recovery guarantees of convex clustering hold only when $\lambda$ satisfies (5). However, in practice, convex clustering is known to perform well even when the condition (5) is not met, e.g., [33] show that exact clustering can be recovered even for values of $\lambda$ not in (5), with, e.g., [32], [34], validating the performance on real data, without the knowledge of (5). We provide numerical experiments that verify these observations in Section 5 ahead.

4 Theoretical guarantees

In this section, we present the theoretical guarantees of the proposed method. For the sake of simplicity, we assume $n_i = n$, for all $i \in [m]$. Subsection 4.1 introduces some intermediate results used in our work. Subsection 4.2 presents the main results of the paper. Subsection 4.3 offers a detailed comparison of our method with the method from [24]. Subsection 4.4 presents the MSE guarantees if the exact solutions of the local empirical risks are replaced by approximate ones.

4.1 Intermediate results

Specializing (5) to the proposed method in Subsection 3.1 we can see that, for the clustering in step 3 of the method proposed in Subsection 3.1 to be correct, we need the following condition satisfied

$$
\max_{k \in [K], i, j \in C_k} \frac{\|\tilde{\theta}_i - \tilde{\theta}_j\|}{|C_k|} \leq \lambda < \min_{k \neq l} \frac{\|\tilde{\theta}_k - \tilde{\theta}_l\|}{2m - |C_k| - |C_l|}. \tag{6}
$$

Next, we state some important results used in the paper, from [9] and [30].

Lemma 1 (Theorem 3 in [30]). Under Assumptions 7, for any $k \in [K]$, any $i \in C_k$ and any $0 < \delta < \frac{1}{2}$, $\epsilon > 0$, with probability at least $1 - 2\delta$, we
have
\[ F_k(\hat{\theta}_k) - F_k(\theta^*_k) \leq \frac{16R^2LC(\epsilon, \delta)}{n} + \frac{8R\|\nabla f_i(\theta^*_k)\| \log \frac{2}{\delta}}{n} + \frac{8LF_k(\theta^*_k) \log \frac{2}{\delta}}{\mu F_k n} + \left(8RL + G_{F_k} + \frac{4RLC(\epsilon, \delta)}{n}\right) \epsilon, \]
where \( C(\epsilon, \delta) := 2 \left( \log \frac{2}{\delta} + d \log \frac{6R}{\epsilon} \right). \)

**Lemma 2** (Theorem 1 in [9]). Under Assumptions 1-5, for each \( k \in [K] \) and \( \bar{\theta}_k = \frac{1}{|C_k|} \sum_{i \in C_k} \hat{\theta}_i \), we have
\[ \mathbb{E}\|\bar{\theta}_k - \theta^*_k\|^2 \leq \frac{2E_k}{n|C_k|} + \frac{5}{\mu^2 F_k n^2} \left( L^2 \log d + E_k \right) + O \left( |C_k|^{-1} n^{-2} \right) + O(n^{-3}), \]
where \( E_k := \mathbb{E}\|\nabla^2 F_k(\theta^*_k)^{-1} \nabla \ell(\theta^*_k; X)\|^2 \).

Note that the original results in [9] and [30] concern the global population loss and the corresponding empirical loss, given as the average loss across all users. These directly translate to each individual cluster in our framework, i.e., to each component in (3). Also, note that Lemma 2 assumes knowledge of the true clusters, as the averaging is performed with respect to the true clusters.

### 4.2 Main results

We are now ready to state the main results of the paper.

**Theorem 1.** Let Assumptions 1-5 hold. If the number of samples per user satisfies \( n \geq 3 \) and \( \frac{n \log n}{\log n} > \frac{2M(2m-|C(K-1)|-|C(K)|)^2}{|C(K)|^2(D-2\gamma)^2} \), where \( \beta \geq 1 \) and \( 0 < \gamma < \frac{D}{2} \) are tunable parameters, \( M = M(\beta) = \max_{i,j \in C_k, k \in [K]} M_{ik} + M_{jk} \), where for \( i \in C_k \), we have \( M_{ik} = \frac{64R^2L(\log 2 + d \log 6R + (d+1)\beta)}{\mu F_k} + \frac{16LF_k(\theta^*_k)\log 2+\beta}{\mu F_k} \)
\[ + \frac{2G_{F_k} + 16RL(1+\log 2+d \log 6R + (d+1)\beta)}{2m-|C(K-1)|} \frac{\mu F_k}{n}, \]
then, for any choice of \( \lambda \in \left[ \sqrt{\frac{2m \log n}{n}}, \frac{|C(K)-(D-2\gamma)}{2m-|C(K-1)|} \right] \), we have that, for all \( k \in [K] \), the
models produced by the proposed method achieve the MSE
\[ \mathbb{E} \| \theta_k - \theta_k^\ast \|^2 \leq \frac{2E_k}{n|C_k|} + \frac{4K\tilde{E}R^2}{n|C(K)|\gamma^2} + \frac{4KR^2|\tilde{C}|^2}{n^3} \]
\[ + \mathcal{O} \left( \frac{\log d}{n^2} \right) + \mathcal{O} \left( \frac{K\log d}{n^2\gamma^2} \right) + \mathcal{O} \left( \frac{1}{n^2|C_k|} \right) \]
\[ + \mathcal{O} \left( \frac{K}{n^2|C(K)|\gamma^2} \right) + \mathcal{O} \left( \frac{1}{n^5} \right) + \mathcal{O} \left( \frac{K}{n^5\gamma^2} \right), \]
where \( E_k = \mathbb{E} \| \nabla^2 F_k(\theta_k^\ast)^{-1} \nabla \ell(\theta_k^\ast; X) \|^2 \), \( \tilde{E} = \frac{1}{K} \sum_{k \in [K]} E_k \) and \( |\tilde{C}|^2 = \frac{1}{K} \sum_{k \in [K]} |C_k|^2 \).

Theorem\[1\] provides the MSE rate of the proposed method. If, in addition to the conditions of Theorem\[1\], we have \( n \geq |C(1)| \), then, for the choice of \( \beta \geq 2 \), we have that the MSE rate is dominated by the first two terms, i.e., \[ \frac{2E_k}{n|C_k|} + \frac{4K\tilde{E}R^2}{n|C(K)|\gamma^2}. \] Since \( 0 < \gamma < \frac{D}{2} \) is a tunable parameter, if \( D > 2 \sqrt{\frac{|C(1)|}{|C(K)|}} \), we can choose \( \gamma = \sqrt{\frac{|C(1)|}{|C(K)|}} \), so that the dominating term becomes \[ \frac{2E_k}{n|C_k|} + \frac{4K\tilde{E}R^2}{n|C(1)|}. \] This observation directly leads to the following corollary.

**Corollary 1.** Let conditions of Theorem \[1\] hold. If additionally \( D > 2 \sqrt{\frac{|C(1)|}{|C(K)|}} \) and \( n \geq |C(1)| \), then, for the choice of \( \beta \geq 2 \), \( \gamma = \sqrt{\frac{|C(1)|}{|C(K)|}} \), the proposed method achieves \( \mathbb{E} \| \theta_k^\ast - \theta_k^\ast \|^2 \leq \mathcal{O} \left( \frac{1}{n|C_k|} \right) \), for all \( k \in [K] \).

Corollary \[1\] shows that, if the populations are sufficiently separated, the method proposed in Subsection \[3.1\] can achieve the order-optimal MSE rate for each cluster, provided that users have sufficient number of samples available. This rate is equivalent to the rate achieved by training a centralized learner on each cluster, while simultaneously achieving significant communication savings, requiring only a single communication round. Some remarks are now in order.

**Remark 7.** The MSE guarantees in Lemma \[2\] are established without any requirements on the sample size \( n \). This stems from the fact that the method from Lemma \[2\] can be seen as an oracle method that knows the true clustering structure. On the other hand, the sample size requirement in Theorem \[7\] stems from the fact that our method does not know the true clustering, hence a sufficiently large sample size that guarantees the true clustering can be recovered, is required.
Remark 8. Recall that the parameters $\beta$, $\gamma$ are tunable. From Theorem 1, we can see that both parameters offer a trade-off between convergence speed and sample requirements, e.g., larger values of $\beta$ and $\gamma$ result in faster convergence, at the expense of higher sample requirements.

Remark 9. Recall the condition on the number of samples in Theorem 1. We can identify three components of the condition that quantify the complexity of different aspects of the system:

1. $M$ - quantifies the difficulty of the learning problems, as it depends on problem parameters, e.g., an easier learning problem implies smaller $M$ and reduced sample requirement;

2. $\frac{2m - |C(K)| - |C(K-1)|}{|C(K)|^2}$ - quantifies how well balanced the clusters are, e.g., when the clusters are well balanced, so that $|C_k| = \frac{m}{K}$, the term evaluates to $4(K-1)^2$, while in the extreme case of $|C(K)| = |C(K-1)| = 1$, we get $4(m-1)^2$;

3. $(D - 2\gamma)^{-2}$ - quantifies the difficulty of the clustering problem, e.g., if $D$ is smaller, different population optima are closer to one another and it is more difficult to cluster the local ERMs correctly, hence requiring more samples.

Remark 10. Recall the condition on $D$ in Corollary 1. $D > 2\sqrt{\frac{|C(1)|}{|C(K)|}}$.

When the clusters are well balanced, so that $|C_k| = \frac{m}{K}$, $k \in [K]$, our method can achieve order-optimal rates if the minimal separation between population optima of different clusters is $D > 2$, i.e., independent of any problem parameters. On the other hand, in the worst case, we can have $D > 2\sqrt{m-1}$, if there are only two clusters $C_1, C_2$, such that $|C_1| = 1, |C_2| = m-1$.

Remark 11. We numerically demonstrate that the MSE guarantees of Theorem 1 can be attained by the method proposed in Subsection 3.2 that uses clusterpath, without requiring any specific interval for $\lambda$, any knowledge on problem parameters, nor the knowledge of $K$. The results can be found in Section 5.

Proof of Theorem 1. We start by noting that, for any event $\Psi$, we have

$$\mathbb{E}\|\hat{\theta}_k - \theta_k^*\|^2 = \mathbb{E}\|\hat{\theta}_k - \theta_k^*\|^2\mathbb{I}_\Psi + \mathbb{E}\|\hat{\theta}_k - \theta_k^*\|^2\mathbb{I}_{\Psi^c},$$

where $\mathbb{I}_\Psi$ is the indicator random variable. We now proceed to define a specific event $\Psi$ and establish the resulting bounds.
Applying Lemma 1 for the choice of $\delta = \epsilon = \frac{1}{n^\beta}$, for some $\beta > 0$, we get that, for all $k \in [K]$ and all $i \in C_k$, we have

$$
\|\hat{\theta}_i - \theta^*_k\|^2 \leq \frac{32R^2LC(\epsilon, \delta)}{n\mu_{F_k}} + \frac{16LF_k(\theta^*_k)(\log 2 + \beta \log n)}{n\mu_{F_k}^2}
+ \frac{16R\|\nabla f_i(\theta^*_k)\|(\log 2 + \beta \log n)}{n\mu_{F_k}}
+ \left(\frac{16RL + 2G_{F_k} + \frac{8RLC(\epsilon, \delta)}{n}}{n^\beta \mu_{F_k}}\right),
$$

with probability at least $1 - \frac{2}{n^\beta}$, where $C(\epsilon, \delta) = 2 \left(\log 2 + d \log 6R + (d + 1)\beta \log n\right)$. Here, we used strong convexity of $F_k$, which implies

$$
\|\hat{\theta}_i - \theta^*_k\|^2 \leq \frac{2}{\mu_{F_k}} \left(F_k(\hat{\theta}_i) - F_k(\theta^*_k)\right).
$$

Note that, for $\beta \geq 1$ and $n \geq 3$, the dominating term in (8), in terms of the number of samples $n$, is of the order $O \left(\frac{\log n}{n}\right)$. We can therefore upper-bound the right-hand side of (8) by $\frac{M_{ik} \log n}{n}$, where $M_{ik}, i \in C_k, k \in [K]$ is defined as follows

$$
M_{ik} = \frac{64R^2L(\log 2 + d \log 6R + (d + 1)\beta)}{\mu_{F_k}}
+ \frac{16LF_k(\theta^*_k)(\log 2 + \beta)}{\mu_{F_k}^2}
+ \frac{16R\|\nabla f_i(\theta^*_k)\|(\log 2 + \beta)}{\mu_{F_k}}
+ \frac{2G_{F_k} + 16RL(1 + \log 2 + d \log 6R + (d + 1)\beta)}{\mu_{F_k}}.
$$

As $\frac{M_{ik} \log n}{n}$ is an upper bound on the right-hand side of (8), we can therefore conclude that, for any $i \in C_k, k \in [K]

$$
\mathbb{P}\left(\|\hat{\theta}_i - \theta^*_k\|^2 \leq \frac{M_{ik} \log n}{n}\right) \geq 1 - \frac{2}{n^\beta}.
$$

(9)

Next, define the events

$$
\Sigma_{ij} = \left\{\omega : \|\hat{\theta}_i - \hat{\theta}_j\|^2 \leq \frac{2(M_{ik} + M_{jk}) \log n}{n}\right\},
$$

$$
\Upsilon_i = \left\{\omega : \|\hat{\theta}_i - \theta^*_k\|^2 \leq \frac{M_{ik} \log n}{n}\right\}.
$$
for all $i, j \in C_k$, $i \neq j$, $k \in [K]$. Noting that
\[
\|\hat{\theta}_i - \hat{\theta}_j\|_2^2 \leq 2\|\hat{\theta}_i - \theta_k^*\|_2^2 + 2\|\hat{\theta}_j - \theta_k^*\|_2^2,
\]
we can conclude that, for all $i, j \in C_k$, $k \in [K]
\[
P(\Upsilon_i \cap \Upsilon_j) \leq P(\Sigma_{ij}). \tag{10}
\]
For $\Sigma = \cap_{i,j \in C_k, i \neq j, k \in [K]} \Sigma_{ij}$, we then get the following bound
\[
P(\Sigma) \geq 1 - \sum_{i \neq j \atop i, j \in C_k, k \in [K]} 1 - \sum_{i \neq j \atop i, j \in C_k, k \in [K]} P((\Upsilon_i \cap \Upsilon_j)^c) \geq 1 - \frac{4}{n^\beta} \geq 1 - \frac{2}{n^\beta} \sum_{k \in [K]} |C_k| (|C_k| - 1)
\]
\[
\geq 1 - \frac{2K|\tilde{C}|^2}{n^\beta},
\]
where $|\tilde{C}|^2 = \frac{1}{K} \sum_{k \in [K]} |C_k|^2$, the first inequality follows from the union bound, the second inequality follows from (10), while the third inequality follows from the union bound and (9). Next, for any $k, l \in [K]$, we have that
\[
\|\bar{\theta}_k - \bar{\theta}_l\| \geq \|\theta_k^* - \theta_l^*\|-\|\bar{\theta}_k - \theta_k^*\|-\|\bar{\theta}_l - \theta_l^*\|.
\tag{11}
\]
For any $\gamma > 0$ and any $k \in [K]$, applying Chebyshev’s inequality and Lemma 2, we get the following bound
\[
P\left(\|\bar{\theta}_k - \theta_k^*\| > \gamma\right) \leq \frac{E\|\bar{\theta}_k - \theta_k^*\|_2^2}{\gamma^2} \leq \frac{2E_k}{n|C_k| \gamma^2}
\]
\[
+ \frac{5 \left(L^2 \log d + E_k\right) E_k}{\mu^2_{\hat{k}} n^2 \gamma^2} + O\left(\frac{1}{|C_k| n^2 \gamma^2}\right) + O\left(\frac{1}{n^3 \gamma^2}\right). \tag{12}
\]
\[17\]
Define the event \( \Lambda = \bigcap_{k \in [K]} \{ \omega : \| \bar{\theta}_k - \theta^*_k \| \leq \gamma \} \). We then have

\[
\mathbb{P}(\Lambda) \geq 1 - \sum_{k \in [K]} \mathbb{P} \left( \| \bar{\theta}_k - \theta^*_k \| > \gamma \right)
\]

\[
\geq 1 - \sum_{k \in [K]} \left( \frac{2E_k}{n|C_k|\gamma^2} + \frac{5}{\mu^2_F n^2 \gamma^2} \frac{(L^2 \log d + E_k) E_k}{\mu^2_F n^2 \gamma^2} \right)
\]

\[
+ \mathcal{O} \left( \frac{1}{C_k n^2 \gamma^2} \right) + \mathcal{O} \left( \frac{1}{n^3 \gamma^2} \right)
\]

\[
\geq 1 - \frac{2K \tilde{E}}{n|C(K)|\gamma^2} - \frac{5K}{\mu^2_F n^2 \gamma^2} \left( \tilde{E} L^2 \log d + \tilde{E}^2 \right)
\]

\[
- \Omega \left( \frac{K}{C(K)|n^2 \gamma^2} \right) - \Omega \left( \frac{K}{n^3 \gamma^2} \right),
\]

where \( \tilde{E} = \frac{1}{K} \sum_{k \in [K]} E_k \), \( \tilde{E}^2 = \frac{1}{K} \sum_{k \in [K]} E_k^2 \) and \( \mu_F = \min_{k \in [K]} \mu_{F_k} \). Recall that \( D = \min_{k \neq l} \| \theta^*_k - \theta^*_l \| \). Applying (11), we then have that on \( \Lambda \), for any \( k, l \in [K] \)

\[
\| \bar{\theta}_k - \bar{\theta}_l \| \geq D - 2\gamma,
\]

which is valid for any \( \gamma < \frac{D}{2} \). Next, notice that on \( \Sigma \), for all \( i, j \in C_k, k \in [K] \), we have

\[
\| \hat{\theta}_i - \hat{\theta}_j \| \leq \sqrt{\frac{2(M_{ik} + M_{jk}) \log n}{n}}.
\]

Plugging (13) and (14) into (6), we get that the true clustering can be recovered if

\[
\sqrt{\frac{2M \log n}{n}} < \frac{|C(K)|D - 2\gamma}{2m - |C(K-1)| - |C(K)|},
\]

where \( M = \max_{i,j \in C_k, k \in [K]} (M_{ik} + M_{jk}) \). For (13) to hold we need the number of samples per user to be such that

\[
\frac{n}{\log n} > \frac{2M(2m - |C(K-1)| - |C(K)|)^2}{|C(K)|^2 (D - 2\gamma)^2}.
\]

We then have that on \( \Psi = \Sigma \cap \Lambda \), if the number of samples per user satisfies (16), the true clustering can be recovered for any choice of \( \lambda \in \left[ \sqrt{\frac{2M \log n}{n}}, \frac{|C(K)|D - 2\gamma}{2m - |C(K-1)| - |C(K)|} \right] \) and Lemma 2 applies to our method. On
the other hand, we have

\[
\mathbb{P}(\Psi) \geq \mathbb{P}(\Sigma) + \mathbb{P}(\Lambda) - 1
\]

\[
\geq 1 - \frac{2K\tilde{E}}{n|C(K)|\gamma^2} - \frac{5K \left( \tilde{E}L^2 \log d + \tilde{E}^2 \right)}{\mu^2_F n^2 \gamma^2} - \Omega \left( \frac{K}{|C(K)|n^2 \gamma^2} \right) - \Omega \left( \frac{K}{n^3 \gamma^2} \right) - \frac{2K|\tilde{C}|^2}{n^3},
\]

which implies

\[
\mathbb{P}(\Psi^c) \leq \frac{2K\tilde{E}}{n|C(K)|\gamma^2} + \frac{2K|\tilde{C}|^2}{n^3} + O \left( \frac{K}{|C(K)|n^2 \gamma^2} \right) + \frac{5K \left( \tilde{E}L^2 \log d + \tilde{E}^2 \right)}{\mu^2_F n^2 \gamma^2} + O \left( \frac{K}{n^3 \gamma^2} \right).
\]

Combining everything in (7), we finally get

\[
\mathbb{E}\| \tilde{\theta}_k - \theta_k^* \|^2 \leq \mathbb{E}\| \tilde{\theta}_k - \bar{\theta}_k^* \|^2 + R^2 \mathbb{P}(\Psi^c) \]

\[
\leq \frac{2E_k}{n|C_k|} + \frac{2K\tilde{E}R^2}{n|C(K)|\gamma^2} + \frac{2KR^2|\tilde{C}|^2}{n^3} + O \left( \frac{1}{|C_k|n^2} \right) + \frac{5E_k}{\mu^2_F n^2} \left( L^2 \log d + E_k \right) + \frac{5R^2K}{\mu^2_F n^2 \gamma^2} \left( \tilde{E}L^2 \log d + \tilde{E}^2 \right)
\]

\[
+ O \left( \frac{K}{|C(K)|n^2 \gamma^2} \right) + O \left( \frac{1}{n^3} \right) + O \left( \frac{K}{n^3 \gamma^2} \right),
\]

for \( n \) satisfying (16).

4.3 Comparison with order-optimal CFL methods

In this section we compare the results from Theorem 1 and Corollary 1 with the guarantees of other CFL methods. We compare our method from Subsection 3.1 of the main paper with two methods, namely Iterative Federated Clustering Algorithm (IFCA) [24] and the method from [25]. Both methods provide statistical optimality guarantees and both are order-optimal, up to logarithmic factors, making for an appropriate comparison. In what follows, due to some similarities of the methods and their performances, we will provide a detailed outline of IFCA, while highlighting where [25] differs significantly.
IFCA is an iterative algorithm for CFL that alternates between the following two steps: inferring cluster membership and updating the models. To that end, IFCA is initialized by first producing $K$ different models $\{\theta_k^0\}_{k \in [K]}$, where the superscript denotes the iteration counter. The method then proceeds as follows, for $t = 0, \ldots, T - 1$:

1. The server broadcasts $\{\theta_k^t\}_{k \in [K]}$ to each user.
2. Each user evaluates the models on their local data and chooses the model $\theta_{(i)}^t$, where $(i) = \arg\min_{k \in [K]} f_i(\theta_k^t)$.
3. Each user computes the local stochastic gradient $g_i^t = \tilde{\nabla} f_i(\theta_{(i)}^t)$, evaluated at $\theta_{(i)}^t$. Users send the gradients back to the server, along with a one-hot encoding vector $s_i \in \mathbb{R}^K$, such that $s_{ij} = \begin{cases} 1, & (i) = j \\ 0, & (i) \neq j \end{cases}$, notifying the server which model was updated by user $i$.
4. The server forms clusters of users that updated specific models, based on the received tokens $\{s_i\}_{i \in [m]}$ and performs the model update, i.e., $\theta_k^{t+1} = \theta_k^t - \alpha \frac{1}{|C_k^t|} \sum_{i \in C_k^t} g_i^t$, where $\alpha > 0$ is the step-size, while $C_k^t = \{i \in [m] : s_{ik} = 1\}$ is the cluster of users that updated model $k$ at iteration $t$.

On the other hand, the method in [25] can be seen as a modular method, as it depends on the following three steps:

1. Each user trains the local ERMs and sends them to the server.
2. The server performs a clustering procedure.
3. A FL algorithm is run on the resulting clusters for $T$ iterations to produce the final models.

From the algorithm above, we can see that both IFCA and the method [25] require knowledge of the true number of distributions $K$ (or at least a good estimate), which is typically unavailable or would require running a separate learning algorithm in practice (e.g., community detection). Secondly, both require $T$ rounds of communication, whereas our method requires a single round of server-user communication.

Assumptions. Similarly to our algorithm, IFCA assumes that the population risks $F_k$, $k \in [K]$ are $L$-smooth and $\mu_{F_k}$-strongly convex. The
method [25] requires a stronger assumption - namely, that the loss function \( \ell \) is strongly convex. Additionally, IFCA assumes bounded variance of \( \ell \) and \( \nabla \ell \) with respect to all \( D_k, k \in [K] \), i.e.,

\[
\mathbb{E}_{X \sim D_k} \left[ (\ell(\theta; X) - F_k(\theta))^2 \right] \leq \eta^2
\]

\[
\mathbb{E}_{X \sim D_k} \left[ \|\nabla \ell(\theta; X) - \nabla F_k(\theta)\|^2 \right] \leq \nu^2,
\]

for some \( \eta > 0, \nu > 0 \). Intuitively, these assumption are made to ensure that the empirical loss \( f_i \) and its gradient \( \nabla f_i \) stay close to the true population loss \( F_k \), enabling clustering inference via the local loss. While our method also relies on the local loss to infer the clustering (via the local ERMs), no such assumptions are required in our paper, instead leveraging statistical learning and strong convexity of \( F_k \) to obtain tight guarantees on the quality of our local estimates. An additional assumption made by IFCA, that is required for the convergence of the algorithm is sufficiently close initialization, i.e., for all \( k \in [K] \), the authors require

\[
\|\theta^0_k - \theta^*_k\| \leq \left( \frac{1}{2} - \alpha_0 \right) D \sqrt{\frac{\mu_F}{L}},
\]

where \( \alpha_0 \in (0, \frac{1}{2}) \) is a tunable parameter that determines the proximity of the initialization to the true population optima. Note that such an assumption is quite strong, as it requires \( \|\theta^0_k - \theta^*_k\| < \frac{1}{2} D \), for all \( k \in [K] \). In order to find such an initialization, the knowledge of underlying clusters, as well as \( D \), would have to be available. The method [25], like our method, does not require such an assumption. IFCA requires three further assumptions:

1. \( |C([K])| \gtrsim \log(mn) \) i.e., the size of the smallest cluster has to grow at least logarithmically in the number of total samples available in the system;

2. \( n \gtrsim \frac{K \eta^2}{\alpha_0 \mu_F D^2} \), i.e., each user contains a sufficient number of samples;

3. \( D \geq \tilde{O} \left( \max \left\{ \alpha_0^{-2/3} n^{-1/3}, \alpha_0^{-1/3} m^{-1/6} n^{-1/3} \right\} \right) \), i.e., the population optimal models across different populations are sufficiently well separated.

\(^3\) Note that the authors in [24] use \( n' = \frac{n}{2T} \) in their theoretical analysis, i.e., they require that each user contains \( n = 2T n' \) samples and all conditions in the original paper are expressed in terms of \( n' \). However, for the sake of simplicity, we will represent the conditions in terms of \( n \), effectively reducing the original sample size requirement by a factor of \( 2T \).
Here, the relation $x \gtrsim y$ indicates the existence of global constant $C$ that does not depend on the problem parameters, such that $x \geq Cy$ (the relation $x \lesssim y$ is defined similarly), while $\tilde{O}(\cdot)$ hides logarithmic factors that do not depend on $m$ and $n$. On the other hand, both our method and the method [25] do not require any assumptions on the separation parameter $D$ or on the size of the smallest cluster $|C_{(K)}|$, effectively covering the cases for which IFCA might fail, i.e., highly unbalanced clusters, e.g., $|C_{(K)}| = \mathcal{O}(1)$ and small separation between optimal models of different populations. Comparing the requirements on the number of samples of our method, IFCA and [25], we have, respectively,

\[
\frac{n}{\log n} > \frac{2M(2m - |C_{(K-1)}| - |C_{(K)}|)^2}{|C_{(K)}|^2(D - 2\gamma)^2},
\]

\[
n \gtrsim \frac{Kn^2}{\alpha_0^2\mu F_D^2},
\]

\[
n \geq \frac{S^2L\log m}{\mu_F^3}.
\]

We can first see that that for our method, the requirement is expressed in terms of $n/\log n$, which, for $n \geq 3$ is slightly more strict than placing requirements directly on $n$, up to a logarithmic factor. This can be explained by the fact that our method performs only one round of communication and estimates the clusters only once, hence a larger sample size is required for the guarantees of Theorem 1 to hold. On the other hand, both IFCA and the method [25] perform multiple rounds of communication, with IFCA also iteratively estimating the clusters, hence producing better cluster estimates as the training progresses. This is also reflected in the dependence on problem constants, as we can see on the right-hand sides of the inequalities. IFCA, the algorithm that performs multiple communication and cluster estimation rounds shows the best dependence in terms of the problem constants. The method [25], that estimates the clusters only once, has a worse dependence compared to IFCA, especially as it depends logarithmically on the number of total users, $m$. As expected, our one-shot method shows the worst dependence on the problem parameters. However, we note that for $D < 1$, our method has a much better dependence on $D$ than IFCA. Moreover, the term $\frac{2m - |C_{(K)}| - |C_{(K-1)}|}{|C_{(K)}|}$ evaluates to $4(K - 1)^2$, for $|C_{(K)}| = \frac{m}{K}$, i.e., balanced clusters. If the number of clusters is much smaller than the total number of users, so that $\log m \geq 4(K - 1)^2$, we achieve a better dependence on the number of samples/clusters than the method [25]. While the higher requirements on the number of samples of our method are to be expected, more importantly,
they uncover regimes in which communicating beyond one round to achieve order-optimality is redundant. Additionally, our method does not require knowledge of \( K \), while both IFCA and [25] assume the knowledge of the true value of \( K \). All of these facts lead to less strict requirements of IFCA on the number of samples per user, with respect to different problem parameters (in some cases).

**Guarantees.** The guarantees of IFCA are given in terms of high probability bounds, while our guarantees, expressed in terms of the MSE, are sharper. IFCA provides the following guarantee (Corollary 2 in [24]): after

\[
T = \frac{8mL}{|C(K)|\mu_F} \log \left( \frac{2D\varepsilon}{\delta} \right)
\]

communication rounds, with probability at least \( 1 - \delta \)

\[
\|\theta^*_k - \theta_k^T\| \leq \varepsilon,
\]

where

\[
\varepsilon \lesssim \frac{vKL\log(mn)(m/|C(K)|)^2}{\mu_F\delta\sqrt{n|C(K)|}} + \Theta \left( \frac{1}{n\sqrt{m}} \right) + \frac{\eta^2L^2(m/|C(K)|)^2K\log(mn)}{\mu_F^2\delta D^4n},
\]

(17)

We can see that, assuming \( n \geq |C(1)| \), with probability at least \( 1 - \delta \), the dominating term in (17) becomes

\[
\|\theta^*_k - \theta_k^T\| = O \left( \frac{\log(mn)}{\delta\sqrt{n|C(K)|}} \right),
\]

for all \( k \in [K] \), which is almost order-optimal, up to a logarithmic factor and dependence on the smallest cluster size. The guarantees of [25] are similar, i.e., via Theorem 1 in [25], we have that: after \( T = O \left( \frac{L + \mu_{F_{\text{max}}} \log \left( \frac{\mu_{F_{\text{max}}}}{2\varepsilon} \right)}{\mu_F} \right) \) communication rounds, where \( \mu_{F_{\text{max}}} = \max_{k \in [K]} \mu_F \), with probability at least \( 1 - \delta \), for all \( k \in [K] \)

\[
\|\theta^*_k - \theta_k^T\| \leq O \left( \frac{\log mn}{\delta\sqrt{n|C_k|}} \right),
\]

where \( \delta \) is specifically chosen in terms of problem parameters in their original paper. On the other hand, from Theorem 4 for \( n \geq |C(1)| \), we have

\[
E \|\overline{\theta}_k - \theta_k^*\| = O \left( \frac{1}{\sqrt{n|C(K)|}} \right),
\]

23
for all $k \in [K]$, which is almost order-optimal, with the dependence on the smallest cluster size. Therefore, we can see that our method removes the logarithmic dependence on the total number of samples, of both IFCA and [25], while simultaneously reducing the communication cost by a factor of $O\left(\frac{\kappa}{p} \log \left(\frac{2D}{\epsilon}\right)\right)$ with respect to IFCA (and similar with respect to [25]), where $\kappa = \frac{L}{\mu_F} \geq 1$ is the condition number, while $p = \frac{|C(K)|}{m} < 1$ is the fraction of users belonging to the smallest cluster, reflecting the difficulty of the clustering problem.

However, we can see that Theorem 1 provides guarantees in terms of the size of the smallest cluster, $|C(K)|$, while Theorem 1 in [25] provides the guarantees in terms of the true cluster size $|C_k|$, for each $k \in [K]$. Applying Corollary 1, for $D > 2\sqrt{|C(1)|/|C(K)|}$, our method matches the dependence on individual cluster sizes of [25], while removing the logarithmic dependence on the total number of samples, thus achieving the order-optimal rate

$$E\|\hat{\theta}_k - \theta_k^*\| = O\left(\frac{1}{\sqrt{n|C_k|}}\right),$$

for all $k \in [K]$, while still providing a reduction in communication cost by a factor of $O\left(\frac{\kappa}{p} \log \left(\frac{2D}{\epsilon}\right)\right)$. Hence, we can see that our method provides order-optimal convergence guarantees, improving on the guarantees of both IFCA and [25] by a factor logarithmic in the total number of samples in the system. Remarkably, this is achieved while simultaneously reducing the communication cost by a factor of $O\left(\frac{\kappa}{p} \log \left(\frac{2D}{\epsilon}\right)\right)$ and without requiring the knowledge of the true number of distributions $K$, while both IFCA and [25] assume knowledge of $K$.

4.4 Inexact ERMs

In this section we consider replacing the ERM model $\hat{\theta}_i = \arg\min_{\theta_i \in \Theta} f_i(\theta_i)$, $i \in [m]$, by an inexact estimate, i.e., an estimate $\tilde{\theta}_i \in \Theta$, such that, with probability at least $1 - \delta$

$$\|\tilde{\theta}_i - \hat{\theta}_i\| \leq \epsilon_\delta, \quad (18)$$

where $\epsilon_\delta$ is a positive value, possibly depending on $\delta$, with $\delta \in (0, 1)$. To that end, we need an additional assumption on the strong convexity of the empirical losses $f_i$, $i \in [m]$. 

24
**Assumption 6.** For all $i \in [m]$ the empirical loss $f_i$ is strongly convex, i.e., there exists a constant $\mu_{f_i} > 0$, such that, for all $\theta, \theta' \in \Theta$, we have

\[
f_i(\theta') \geq f_i(\theta) + \langle \nabla f_i(\theta), \theta' - \theta \rangle + \frac{\mu_{f_i}}{2} \| \theta - \theta' \|^2.
\]

Denote by $\mu_f = \min_{i \in [m]} \mu_{f_i}$.

**Remark 12.** Note that in general, Assumption 6 allows for the loss function $\ell$ to be convex, as long as the average across local samples, $f_i(\theta) = \frac{1}{n} \sum_{j=1}^{n} \ell(\theta; x_{ij})$, is strongly convex.

Assumption 6 allows for each user to apply iterative solvers, to obtain parameters $\widehat{\theta_i}$ that satisfy (18). A standard choice is the (projected) stochastic gradient descent (SGD) algorithm \cite{39}. SGD follows a simple update rule, given by

\[
\theta^{t+1} = \Pi_\Theta \left( \theta^t - \eta^t g^t \right),
\]

(19)

where $\theta^t$ is the estimate of the parameter of interest at iteration $t$, $\eta^t$ is the step-size, $g^t$ is a stochastic gradient evaluated at $\theta^t$, with $\Pi_\Theta (\cdot)$ the projection operator onto the set $\Theta$.

SGD can be implemented in both the online setting, where users only have access to a single stochastic gradient at a time and in the batch setting, where users have access to the entire local dataset. Additionally, SGD offers the most general guarantees with respect to the mini-batch size and can be implemented even with a mini-batch size of 1. We discuss at the end of the section how different assumptions can allow for the implementation of more efficient algorithms, in terms of the local iteration complexity per user. Next, we state an additional assumption on the stochastic gradients of $f_i$.

**Assumption 7.** For each $i \in [m]$ and all $\theta \in \Theta$, stochastic gradient $g_i$ of $f_i$, evaluated at $\theta$, are unbiased, i.e., $\mathbb{E}[g_i] = \nabla f_i(\theta)$. Additionally, the stochastic gradients have bounded variance, i.e., there exists a $\sigma_i > 0$, such that for all $\theta \in \Theta$, we have

\[
\mathbb{E}\|g_i - \nabla f_i(\theta)\|^2 \leq \sigma_i^2.
\]

**Remark 13.** Assumption 7 is standard in the analysis of stochastic algorithms, e.g., \cite{40, 41, 42}.

**Remark 14.** Recall the discussion in Section 2 about the boundedness of the gradients of $f_i$, with constants $G_{f_i}$, $i \in [m]$. Combining with Assumption 7 it then follows that, for all $\theta \in \Theta$

\[
\mathbb{E}\|g_i\|^2 = \mathbb{E}\|g_i - \nabla f_i(\theta)\|^2 + \|\nabla f_i(\theta)\|^2 \leq \sigma_i^2 + G_{f_i}^2.
\]
Define $\Gamma^2_i := \sigma^2_i + G^2_i$, $i \in [m]$ and denote by $\Gamma^2 = \max_{i \in [m]} \Gamma^2_i$. We now state two well-known results on the convergence of SGD from [40], used in the rest of the section.

**Lemma 3** (Lemma 1 in [40]). Under Assumptions 2, 3, 6 and 7, for all $i \in [m]$, if we set the step-size rule of SGD as $\eta^t_i = \frac{1}{\mu^t_i}$, it holds for any $T \geq 1$ and any $i \in [m]$ that

$$E\|\theta^T_i - \hat{\theta}_i\|^2 \leq \frac{4\Gamma^2_i}{\mu^2_i T}.$$ 

**Lemma 4** (Lemma 2 in [40]). Let Assumptions 2, 3, 6 and 7 hold and let the stochastic gradients satisfy $\|g^t_i\|^2 \leq \Gamma^2$, with probability 1, for all $i \in [m]$. Then, for all $i \in [m]$ and any $\delta \in (0, 1/e)$, $T \geq 4$, if we set the step-size rule of SGD as $\eta^t_i = \frac{1}{\mu^t_i}$, it holds with probability $1 - \delta$, for any $t \in \{8, \ldots, T - 1, T\}$ and any $i \in [m]$, that

$$\|\theta^t_i - \hat{\theta}_i\|^2 \leq \frac{12\Gamma^2}{\mu^2_i T} + 8\Gamma(121\Gamma + 1)\log \left(\frac{\log(t/\delta)}{t}\right).$$

**Remark 15.** Note that Lemma 4 requires uniformly bounded stochastic gradients almost surely, i.e., $\|g^t_i\|^2 \leq \Gamma^2$, with probability 1, for all $i \in [m]$, which in turns implies uniformly bounded gradient noise. This assumption can be relaxed to allow for sub-Gaussian noise, e.g., using the non-uniformly averaged SGD from [43]. The algorithm performs the same update as the standard SGD, given by (19), with the only difference being in the final estimator, as the SGD simply returns the last iterate $\theta^T_i$, with the non-uniformly averaged SGD from [43] returning

$$\overline{\theta}^T_i = \sum_{t=1}^{T} \frac{t}{T(T+1)/2} \theta^t_i.$$ 

Theorem C.3. in [43] then shows the following guarantee for such an estimator

$$\|\overline{\theta}^T_i - \hat{\theta}_i\|^2 = O\left(\frac{\log (1/\delta)}{T}\right),$$

which improves the results of Lemma 4 by a factor of $\log \log T$. However, due to the ease of implementation of the algorithm from [40] (simply outputing the final iterate $\theta^T_i$), we present the results of the standard SGD approach.

We are now ready to state and prove counterparts of Lemmas 1 and 2 when an inexact ERM estimator is used.
Lemma 5. Let Assumptions 1-4, 6, 7 hold and let the stochastic gradients satisfy \( \|g_i\|^2 \leq \Gamma^2 \), with probability 1, for all \( i \in [m] \). If each user runs SGD locally for \( T \) iterations, with the step-size rule \( \eta_t = \frac{1}{\mu_f} \), to produce \( \theta_i^T \), \( i \in [m] \) and \( T \) is chosen such that \( T \geq 15 \) and \( T \log \log(T) \geq \left( \frac{12\Gamma^2}{\mu_f^2} + 8\Gamma(121\Gamma + 1)(1 + \log \frac{1}{\delta}) \right) \frac{1}{\epsilon^2} \), then for any \( k \in [K] \), any \( i \in C_k \) and any \( \epsilon > 0 \), \( 0 < \delta < \frac{1}{3} \), with probability at least \( 1 - 3\delta \), we have, for any \( i \in C_k \), \( k \in [K] \)

\[
F_k(\theta_i^T) - F_k(\theta^*_k) \leq \frac{16R^2LC(\epsilon, \delta)}{n} + \frac{8R\|\nabla f_i(\theta^*_k)\| \log \frac{2}{\delta}}{n} + \frac{8LF_k(\theta^*_k) \log \frac{2}{\delta}}{\mu F_k n} + \left( 8RL + G_{F_k} + \frac{4RLC(\epsilon, \delta)}{n} \right) \epsilon + \varepsilon S,
\]

where \( C(\epsilon, \delta) := 2 \left( \log \frac{2}{\delta} + d \log \frac{6R}{\epsilon} \right) \).

Proof. For any \( \theta \in \Theta \), any \( k \in [K] \) and any \( i \in C_k \), we have

\[
F_k(\theta) - F_k(\theta^*_k) \leq |F_k(\theta) - F_k(\hat{\theta}_i)| + F_k(\hat{\theta}_i) - F_k(\theta^*_k). \tag{20}
\]

We can bound the second term on the right hand side of (20) using Lemma 1. To bound the first term, we use Lipschitz continuity of \( F_k \) (recall the discussion in Section 2), to get

\[
|F_k(\theta) - F_k(\hat{\theta}_i)| \leq G_{F_k} \|\theta - \hat{\theta}_i\|. \tag{21}
\]

Next, applying Lemma 4, we have that, with probability at least \( 1 - \delta \)

\[
\|\theta_i^T - \hat{\theta}_i\|^2 \leq \frac{12\Gamma^2}{\mu_f^2 T} + 8\Gamma(121\Gamma + 1) \frac{\log (\log(T)/\delta)}{T}.
\]

As \( T \geq 15 \), we can use the following upper-bound, with probability at least \( 1 - \delta \)

\[
\|\theta_i^T - \hat{\theta}_i\|^2 \leq \frac{12\Gamma^2 \log \log(T)}{\mu_f^2 T} + 8\Gamma(121\Gamma + 1) \left( 1 + \log \frac{1}{\delta} \right) \frac{\log \log(T)}{T}.
\]

27
From the conditions of the Lemma, we can then conclude that
\[ \|\theta^T_i - \hat{\theta}_i\| \leq \varepsilon. \] (22)
Plugging (22) into (21) and combining in (20), we finally get that, with probability at least \(1 - \delta\)
\[ F_k(\theta_i^T) - F_k(\theta_i^*) \leq \varepsilon G_F + F_k(\hat{\theta}_i) - F_k(\theta_i^*). \]
The result is completed by applying Lemma 1 to the second term on the right hand side of the final inequality.

**Lemma 6.** Let Assumptions 1-7 hold and each user runs SGD locally for \(T\) iterations, to produce \(\theta_i^T\). If \(T \geq \frac{4\Gamma^2}{\mu_F^2} \varepsilon\), then for \(\tilde{\theta}_k = \frac{1}{|C_k|} \sum_{i \in C_k} \theta_i^T\), \(k \in [K]\), we have
\[ \mathbb{E}\|\tilde{\theta}_k - \theta_i^*\|^2 \leq \frac{4E_k}{|C_k|} + \frac{10}{\mu_F^2 n^2} (L^2 \log d + E_k) E_k \]
\[ + \mathcal{O}\left(|C_k|^{-1} n^{-2}\right) + \mathcal{O}(n^{-3}) + \varepsilon, \]
where \(E_k := \mathbb{E}\|\nabla^2 F_k(\theta_i^*)^{-1} \nabla \ell(\theta_i^*; X)\|^2\).

**Proof.** From Lemma 3, we know that, for each \(i \in [m]\), running SGD locally for \(T \geq \frac{4\Gamma^2}{\mu_F^2} \varepsilon\) iterations results in
\[ \mathbb{E}\|\theta_i^T - \hat{\theta}_i\|^2 \leq \varepsilon. \] (23)
Define the across-cluster average of \(\varepsilon\)-inexact approximations as \(\tilde{\theta}_k = \frac{1}{|C_k|} \sum_{i \in C_k} \theta_i^T\).
We then have
\[ \mathbb{E}\|\tilde{\theta}_k - \theta_i^*\|^2 \leq 2\mathbb{E}\|\bar{\theta}_k - \theta_i^*\|^2 + 2\mathbb{E}\|\bar{\theta}_k - \tilde{\theta}_k\|^2, \] (24)
where \(\bar{\theta}_k = \frac{1}{|C_k|} \sum_{i \in C_k} \bar{\theta}_i\). We can bound the first term on the right-hand side of (24) using Lemma 2. For the second term, we use (23), to obtain
\[ \mathbb{E}\|\bar{\theta}_k - \tilde{\theta}_k\|^2 \leq \frac{1}{|C_k|} \sum_{i \in C_k} \mathbb{E}\|\theta_i^T - \hat{\theta}_i\|^2 = \varepsilon. \]
Combining the results and plugging in (24), we get
\[ \mathbb{E}\|\tilde{\theta}_k - \theta_i^*\|^2 \leq \frac{4E_k}{|C_k|} + \frac{10}{\mu_F^2 n^2} (L^2 \log d + E_k) E_k \]
\[ + \mathcal{O}\left(|C_k|^{-1} n^{-2}\right) + \mathcal{O}(n^{-3}) + \varepsilon, \]
which completes the proof. \(\square\)
Lemmas 5 and 6 give us the counterparts of Lemmas 1 and 2 in the case where an approximate solution to the ERM is used instead of the exact one. We can apply them to prove the following.

**Theorem 2.** Let Assumptions 1-7 hold and let the stochastic gradients satisfy \( \|g^i_t\|^2 \leq \Gamma^2 \) with probability 1, for all \( i \in [m] \). If each user runs SGD locally for \( T \) iterations to produce \( \theta^T_i \), \( i \in [m] \) and the number of samples per user \( n \) and the number of local iterations \( T \) are such that

\[
T \geq \max \left( 15, \frac{41^2}{\mu^2} \right) \quad \text{and moreover}
\]

\[
\frac{n}{\log n} > 2M \left( \frac{(D - 2\gamma)^2|C(K)|^2}{(2m - |C(K)| - |C(K-1)|)^2} - 4\varepsilon \tau F \right)^{-1},
\]

\[
\frac{T}{\log \log (T)} \geq \left( \frac{12\Gamma^2}{\mu^2} + 8\Gamma(12\Gamma + 1)(1 + \beta \log n) \right) \frac{1}{\varepsilon^2},
\]

where \( \beta \geq 1 \) and \( 0 < \gamma < \frac{D}{2} \) are tunable parameters, \( \tau F = \max_{k \in [K]} \frac{G_{F_k}}{\mu_{F_k}} \), while \( M = M(\beta) = \max_{i,j \in C_k, k \in [K]} M_{ik} + M_{jk} \), and for all \( i \in C_k, k \in [K] \)

\[
M_{ik} = 64R^2L \left( \log 2 + d \log 6R + (d + 1)\beta \right) \frac{\mu_{F_k}}{\mu_{F_k}} + \frac{16L F_k(\theta^*_k)(\log 2 + \beta)}{\mu_{F_k}} + \frac{16R\|\nabla f_i(\theta^*_k)\|}{\mu_{F_k}} (\log 2 + \beta) + \frac{2G_{F_k}}{\mu_{F_k}} + 16RL \left( 1 + \log 2 + d \log 6R + (d + 1)\beta \right),
\]

then, for any choice of \( \lambda \in \left[ \sqrt{\frac{2M \log n}{n} + 4\varepsilon \tau F}, \frac{|C(K)|(D - 2\gamma)}{2m - |C(K)| - |C(K-1)|} \right] \), we have that, for all \( k \in [K] \), the models produced by the inexact method achieve the MSE

\[
\mathbb{E}\|\tilde{\theta}_k - \theta^*_k\|^2 \leq \frac{4E_k}{n|C_k|} + \frac{4K \tilde{E}R^2}{n|C(K)|\gamma^2} + \frac{3K R^2|\bar{C}|^2}{n^2} + \frac{10E_k}{\mu_{F_k}^2 n^2} \left( L^2 \log d + E_k \right) + \frac{10R^2K}{\mu_{F_k}^2 n^2 \gamma^2} \left( \tilde{E}L^2 \log d + \tilde{E}^2 \right) + \mathcal{O} \left( \frac{1}{|C_k| n^2} \right) + \mathcal{O} \left( \frac{K}{|C(K)| n^2 \gamma^2} \right) + \mathcal{O} \left( \frac{1}{n^3} \right) + \mathcal{O} \left( \frac{K}{n^3 \gamma^2} \right) + \varepsilon \left( 1 + \frac{2R^2K}{\gamma^2} \right),
\]

29
where $E_k = \mathbb{E}\|\nabla^2 F_k(\theta^*_k)^{-1}\nabla \ell(\theta^*_k; X)\|^2$, $\bar{E} = \frac{1}{K} \sum_{k \in [K]} E_k$, $\bar{E}^2 = \frac{1}{K} \sum_{k \in [K]} E_k^2$ and $|\bar{C}|^2 = \frac{1}{K} \sum_{k \in [K]} |C_k|^2$.

We can provide an analogue to Corollary 1 in the inexact ERM scenario.

**Corollary 2.** Let conditions of Theorem 2 hold. If additionally $D > 2 \sqrt{\frac{|C(1)|}{|C(K)|}}$ and $n \geq |C(1)|$, then for the choices of $\beta \geq 2$ and $\gamma = \sqrt{\frac{|C(1)|}{|C(K)|}}$, we have the following MSE, for all $k \in [K]$

$$\mathbb{E}\|\bar{\theta}_k - \theta^*_k\|^2 \leq O \left( \frac{1}{n|C_k|} + \varepsilon \right) .$$

The proof of Theorem 2 follows the same idea as the proof of Theorem 1, replacing the results of Lemmas 1 and 2 with results from Lemmas 5 and 6. For the sake of brevity, we omit the proof. Some comments are now in order.

**Remark 16.** Note that the sample size requirement implicitly places a requirement on the precision up to which we solve the local ERMs, i.e., we have

$$\varepsilon < \frac{(D - 2\gamma)^2 |C(K)|^2}{4\tau_F(2m - |C(K)| - |C(K-1)|)^2}.$$ 

This requirement can again be seen in terms of the "problem difficulty", with respect to different system aspects. For example, if the clusters are well separated, so that $D-2\gamma$ is large, we can solve the local ERMs up to moderate, or even low precision, while for clusters that are not well separated, we need to solve the local ERMs to high precision in order to achieve the optimal rates. Similarly, if the clusters are well balanced, i.e., $|C_k| = \frac{m}{K}$, for all $k \in [K]$, the term $\frac{|C(K)|^2}{(2m-|C(K)|-|C(K-1)|)^2}$ evaluates to $\frac{1}{4(K-1)^2}$, while in the extreme case of $|C(K)| = |C(K-1)| = 1$, the term evaluates to $\frac{1}{4(m-1)^2}$. For $K \ll m$, we see that balanced clusters (easier clustering problem) again lead to a lower precision requirement than the imbalanced clusters case. Finally, recall that $\tau_F = \max_{k \in K} G_{F_k}/\mu_{F_k}$, where $G_{F_k}$ and $\mu_{F_k}$ are the Lipschitz continuity and strong convexity constants of $F_k$’s, respectively, hence showing that, if $F_k$’s are better behaved (higher $\mu_{F_k}$ and lower $G_{F_k}$), the overall precision to which we have to solve the local ERMs is relaxed.

**Remark 17.** Comparing the MSE rates of Theorem 1 and Theorem 2, we can see that the main difference is the presence of an additional term in
Theorem 2, that being
\[ \varepsilon \left( 1 + \frac{2R^2K}{\gamma^2} \right), \]
with \( \varepsilon > 0 \) representing the accuracy up to which we solve the local ERM. We can therefore see that, as long as the local ERMs are solved up to precision
\[ \varepsilon < \min \left\{ \frac{1}{n|C(1)|}, \frac{(D - 2\gamma)^2|C(K)|^2}{4\tau_F(2m - |C(K)| - |C(K-1)|)^2} \right\}, \]
the rates of Theorem 1 are recovered, i.e., the final MSE is of the order \( O\left( \frac{1}{n|C(K)|} \right) \), for all \( k \in [K] \). This in turns leads to a local iteration requirement of \( T \geq \max \left\{ 15, \frac{4\tau^2}{\mu F} \right\} \) and
\[ \frac{T}{\log \log (T)} \geq \left( \frac{6L\Gamma^2}{\mu F^2} + 4L\Gamma(12\Gamma + 1)(1 + \beta \log n) \right) \frac{1}{\varepsilon^2}, \]
with \( \varepsilon \) satisfying (25).

Remark 18. We can see from Corollary 2 that, if the precision up to which we solve the local problems, \( \varepsilon \), satisfies (25), we again obtain the order-optimal MSE rates
\[ \mathbb{E}\| \tilde{\theta}_k - \theta^*_k \|^2 = O\left( \frac{1}{n|C_k|} \right), \]
for all \( k \in [K] \).

Remark 19. The choice of SGD as the local solver is based on the flexibility offered by the algorithm. The results from Lemmas 3 and 4 do not depend on either the setting being online or locally stored data, nor do they place any requirement on the mini-batch size used. This however leads to sub-optimal dependence on \( \varepsilon \) in the requirements on the number of local iterations each user has to run.

Remark 20. If all the \( n \) local data samples were available to each user, variance reduction methods such as SAGA [41] and SVRG [42] could be applied, making the number of iterations \( T \) dependence on \( \varepsilon \) only logarithmical, i.e., \( T = O\left( \log \frac{1}{\varepsilon} \right) \).

Remark 21. Finally, we remark that Assumption 7 is the most general form assumption on the loss function and as such, leads to the requirement of solving the ERM to precision \( \varepsilon^2 \). As shown in [28], Theorem 2, if the loss is a
generalized linear loss, then it suffices to solve the ERM up to precision \( \varepsilon \). While such an assumption is satisfied by a certain class of strongly convex loss functions, such as \( \ell^2 \) regularized support vector machines, \( \ell^2 \) regularized logistic regression, as well as linear regression, it is less general than Assumption \( \ref{assumption:strong_convexity} \).

5 Numerical experiments

In this section we present numerical experiments. In Subsection \( \ref{subsection:linear_regression} \) we present experiments on linear regression. In Subsection \( \ref{subsection:logistic_regression} \) we present experiments on logistic regression. In Subsection \( \ref{subsection:clusterpath} \) we evaluate the performance of our method using clusterpath, introduced in Subsection \( \ref{subsection:clusterpath} \). In Subsection \( \ref{subsection:ifca} \) we compare the performance of our method with IFCA. All experiments were implemented in python. To solve the local empirical risk problems and convex clustering, we use CVXPY \( \cite{diamond2016cvxpy} \).

5.1 Linear regression

In this subsection we consider a linear regression problem. The data generating process for each cluster is given by

\[
y = \langle x, u^*_k \rangle + \epsilon,
\]

where \( \epsilon \sim \mathcal{N}(0, 1) \), i.e., \( \epsilon \) follows a standard Gaussian distribution, while \( (x, y) \) are pairs of regressors (input variables) and observed values (response variables), respectively. The number of clusters is set to \( K = 10 \). The vectors \( u^*_k \) are \( d \)-dimensional, with \( d = 20 \), and each component is drawn from a uniform distribution, independent of one another. Specifically, we drew \( u^*_k \)'s as:

\[
u^*_1 \sim U([1, 2]), \quad u^*_2 \sim U([4, 5]), \quad u^*_3 \sim U([7, 8]), \quad u^*_4 \sim U([10, 11]), \quad u^*_5 \sim U([13, 14]), \quad \text{with } u^*_6 \text{ through } u^*_10 \text{ begin generated from the corresponding negative intervals},
\]

\[
i.e., \quad u^*_6 \sim U([-2, -1]), \quad \text{through to } u^*_10 \sim U([-14, -13]), \quad \text{respectively, for all } i \in [d].
\]

Such a choice of \( u^*_k \)'s ensures that \( D > 0 \). Each cluster is assigned a total of \( N_k = 100000 \) points, where the datapoints \( x \) are generated as follows: for each \( x \in \mathbb{R}^d \), we choose 5 random components in \([d] \) that are distributed according to \( \mathcal{N}(0, 1) \), while the other components are set to zero. A similar setup was considered in \( \cite{goldfarb2012fast} \), with \( K = 1 \).

To measure the error, we use the quadratic loss, i.e.,

\[
\ell((x, y); u) = (y - \langle x, u \rangle)^2.
\]

Under the proposed loss, we have that \( u^*_k \)'s are the population optimal models, i.e., \( u^*_k = \arg \min_u F_k(u), \ k \in [K] \).
We consider a FL system with $m = 100$ users and a balanced clustering, i.e., $|C_k| = \frac{m}{K} = 10$, for all $k \in [K]$. Each user $i \in C_k$ is assigned $n$ points uniformly at random, from the corresponding sample $N_k$, such that no data point is assigned to two different users, effectively simulating an IID distribution of data within clusters. We benchmark the proposed method with the following methods:

- **Oracle Averaging** - an oracle method that knows the true clusters beforehand and applies the averaging method from [9] on each individual cluster, i.e.,
  \[
  \bar{u}_k = \frac{1}{|C_k|} \sum_{i \in C_k} \hat{u}_i, \tag{26}
  \]
  with $\hat{u}_i$ the local ERM of user $i$ and $C_k, k \in [K]$ being the true underlying clustering;

- **Cluster Oracle** - an oracle method that contains all of the data points assigned to the users from the same clusters, i.e., a total of $\frac{mn}{K}$ data points per cluster and trains the models on all of the data, i.e.,
  \[
  u_k = \arg \min_u \frac{K}{m} \sum_{i \in C_k} f_i(u),
  \]
  with $f_i$’s given by (1);

- **Local ERMs** - ERMs trained on each user’s local data;

- **Naive averaging** - the method from [9], that averages the local ERMs across all users, oblivious to system heterogeneity.

Cluster Oracle is the equivalent of centralized learning, i.e., is the method that trains on all the data available in the cluster, achieving the best order-optimal MSE rate $O\left(\frac{1}{n|C_k|}\right)$ (e.g., [28]). On the other hand, [9] show that Oracle Averaging matches the performance of Cluster Oracle if the sample size is above a threshold. Therefore, using Cluster Oracle and Oracle Averaging as benchmarks illustrates: 1) how fast our method attains the order-optimal MSE rate and 2) the additional requirements on the sample size to reach the order-optimal rate, compared to Oracle Averaging, that stem from not knowing the true clustering.

To measure the quality of performance, we present the average normalized MSE, i.e., for each of the above estimators, we compute
\[
\frac{1}{m} \sum_{i=1}^{m} \frac{\|\bar{u}_i - u^*_i\|^2}{\|u^*_i\|^2}, \tag{27}
\]
where $u^*_i$ denotes the population optima associated with user $i$, while $\tilde{u}_i$ is the estimator associated with user $i$. For example, if we measure the performance of Oracle Averaging estimator from (26), (27) evaluates to
\[
\frac{1}{K} \sum_{k \in [K]} \frac{\|\tilde{u}_k - u^*_k\|^2}{\|u^*_k\|^2}.
\]

To select the parameter $\lambda$, we first compute the lower and upper bounds in (6). If the condition is satisfied, so that the lower bound is strictly smaller than the upper bound, we choose $\lambda$ uniformly at random from the interval defined by the lower and upper bounds in (6). Otherwise, for simplicity, we take lambda to be equal to the upper bound. All results presented below were averaged across 20 runs.

Figure 1: Left: Performance of different methods for linear regression, versus the number of samples available per user. We can see that the proposed method matches the order-optimal MSE rates for a sufficiently large sample size.

Right: Number of clusters produced by convex clustering for linear regression, versus the number of samples available per user. We can see that convex clustering is able to recover the exact clustering for a sufficiently large sample size available to each user.

Figure 1 presents the performance using linear regression models. The left figure presents the MSE of different methods, while the right figure presents the number of clusters produced by convex clustering. On $y$-axis in the left figure we plot the averaged normalized MSE, while on $x$-axis, we present the number of samples $n$ available to each user. We can see that, for a small number of samples (less than 300), the method proposed in Subsection 3.1 clusters each user to an individual cluster, effectively performing like the local ERM. This can be explained by the fact that in the small sample size regime, the conditions of Theorem 1 are not satisfied (with high probability) and the interval for the optimal choice of $\lambda$ will be empty. In such
settings, the upper bound will typically be small, hence resulting in a large number of clusters. On the other hand, as \( n \) grows, we see a sharp phase transition in the quality of our estimator, after which the performance of the proposed method matches the order-optimal performance of both the oracle methods, as predicted by the theory. The difference in the number of samples required for reaching order-optimal rates of our proposed method and the Oracle Averaging (450 and 350 samples required, respectively), as outlined above, stems from the additional requirements of the proposed method to produce the exact clustering. Finally, we see that Naive averaging consistently performs badly, as it is completely oblivious to the clustering structure, illustrating that a global model can be bad in the presence of system heterogeneity. On \( y \)-axis in the right figure we plot the number of clusters produced by the convex clustering algorithm. On \( x \)-axis we again plot the number of samples \( n \). The right figure is consistent with the results from the left figure, as it shows that, for small \( n \) (less than 300), convex clustering clusters each user separately, which, due to the low sample regime and our sub-optimal choice of \( \lambda \), is to be expected. On the other hand, there is a sharp phase transition in the number of clusters for \( n \) between 300 and 400, after which convex clustering consistently produces \( K' = 10 \) clusters. Moreover, we can see that the clustering produced by the convex clustering method is correct, as our method matches the performance of both oracle methods that know the true clustering.

5.2 Logistic regression

In this subsection, we consider a logistic regression problem. The logistic regression model assumes that the data is generated as follows: for each pair \((x_{ij}, y_{ij})_{j=1}^n, i \in C_k\), the label \( y_{ij} \) is generated according to \( y_{ij} = 2\text{Bernoulli}(p_{ij}) - 1 \), where \( \text{Bernoulli}(p_{ij}) \) is a sample from the Bernoulli distribution with parameter \( p_{ij} \). This leads to \( y_{ij} \in \{\pm 1\} \), while we compute the probabilities \( p_{ij} \) by

\[
p_{ij} = \frac{1}{1 + \exp\left(-\left(\langle x_{ij}, \theta^*_k \rangle + b^*_k \right)\right)}.
\]

The number of clusters is set to \( K = 4 \). The vectors \( \theta^*_k \) are \( d \)-dimensional, with \( d = 2 \) (corresponding to the weight parameters) with \( b^*_k \) a scalar (corresponding to the intercept). Specifically, we chose \( \theta^*_k \)'s as: \( \theta^*_1 = [1 \ -1]^T \), \( \theta^*_2 = [1 \ 0]^T \), \( \theta^*_3 = [-1 \ 1]^T \) and \( \theta^*_4 = [0 \ -1]^T \), with \( b^*_k = 0 \), for all \( k \in [K] \). Such a choice of \( \theta^*_k \)'s ensures that \( D > 0 \). Each cluster is assigned a total of \( N_k = 100000 \) points, with the datapoints \( x \) all centered
at $\mu = [0 \ 0]^T$, with covariance matrices $\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\Sigma_2 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$, $\Sigma_3 = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$ and $\Sigma_4 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$, corresponding to clusters $C_1$ through to $C_4$, respectively.

To measure the error, we use the $\ell_2$ regularized logistic loss, i.e.,

$$\ell((x, y); \theta, b) = \log(1 + \exp(-y(\langle x, \theta \rangle + b))) + \frac{C\|\theta\|^2}{2},$$

where $C > 0$ is a regularization parameter. In our experiments, we set $C = 10^{-5}$. Under the proposed loss, we have that $\theta^*_k$’s are the population optimal models, i.e., $\theta^*_k = \arg \min_u F_k(u), k \in [K]$.

We consider a FL system with $m = 100$ users and a balanced clustering, i.e., $|C_k| = \frac{m}{K} = 25$, for all $k \in [K]$. Each user $i \in C_k$ is assigned $n$ points uniformly at random, from the corresponding sample $N_k$, such that no data point is assigned to two different users, again simulating an IID distribution of data within clusters. We use the same benchmarks, error metric and selection procedure for $\lambda$, as described in Section 5 of the main paper. All results were averaged across 20 runs.

Figure 2: Left: Performance of different methods for logistic regression, versus the number of samples available per user. We can see that the proposed method matches the order-optimal MSE rates for a sufficiently large sample size. Right: Number of clusters produced by convex clustering for logistic regression, versus the number of samples available per user. We can see that convex clustering is able to recover the exact clustering for a sufficiently large sample size available to each user.

Figure 2 presents the performance using logistic regression models. The left figure presents the MSE of different methods, while the right figure presents the number of clusters produced by convex clustering. On the $y$-axis in the left figure we plot the averaged normalized MSE, while on the
x-axis we present the number of samples $n$ available to each user. We can see that, for a small number of samples (less than 500), the proposed method (Subsection 3.1 of the main paper) clusters each user to an individual cluster, effectively performing like the local ERM. As $n$ grows, we see that the quality of our estimator increases, eventually matching the performance of the order-optimal oracle methods (for $n \geq 4600$), as predicted by the theory. The difference in the number of samples required for reaching order-optimal rates of our proposed method and the Oracle Averaging is much larger compared to the linear regression problem. This can be explained by the fact that the sample requirements of Oracle Averaging depend only on the size of the largest cluster, while the sample requirements of the proposed method depend on the problem parameters via the value $M$ (recall Theorem 1), as well as the size of the largest cluster. As logistic regression is a more complex and more “nonlinear” model compared to linear regression, the sample requirements of the proposed method in Subsection 3.1 of the main paper increase accordingly. On y-axis in the right figure we plot the number of clusters produced by the convex clustering algorithm. On x-axis we again plot the number of samples $n$. The right figure is consistent with the results from the left figure, as it shows that, for small $n$ (less than 500), convex clustering clusters each user separately, which, due to the low sample regime and our sub-optimal choice of $\lambda$, is to be expected. As the number of samples grows, convex clustering produces lower number of clusters. Zooming in on the interval $3000 \leq n \leq 5000$ (subplot in right figure), we can see that after reaching 3000 samples, convex clustering will on average produce $K' = 4$ models (consistent with the ground truth), with some variations. This is again consistent with the figure on the left of Figure 2, as we see that for larger $n$ our method is getting closer to the performance of the two oracle methods, exactly matching them for $n \geq 4600$, when, as we can see from the figure on the right, convex clustering is consistently producing $K' = 4$ clusters. Again, we can see that the clustering produced by the convex clustering method is correct, as our method closes in on the performance of both oracle methods that know the true clustering, eventually matching them for $n$ large enough.

5.3 Clusterpath

In this subsection, we evaluate the performance of the method proposed in Subsection 3.2 of the main paper, that uses clusterpath, and compare it to the performance of the method proposed in Subsection 3.1 of the main paper. We consider a linear regression problem, with $K = 4$. The experiment
design is identical to the one in Section 5 of the main body, with the optimal models chosen as $u^*_{1i} \sim U([0, 1])$, $u^*_{2i} \sim U([1, 2])$, $u^*_{3i} \sim U([-1, 0])$ and $u^*_{4i} \sim U([-2, -1])$, $i = 1, \ldots, d$, with $d = 20$.

We again consider a FL system with $m = 100$ users and a balanced clustering, i.e., $|C_k| = \frac{m}{K} = 10$, for all $k \in [K]$. Each user $i \in C_k$ is assigned $n$ points uniformly at random, from the corresponding sample $N_k$, such that no data point is assigned to two different users. We compare the following two methods:

- **Exact convex clustering** - the method proposed in Subsection 3.1 in the main paper;
- **Clusterpath** - the method proposed in Subsection 3.2 in the main paper.

To find $\lambda_1$ such that $K_{\lambda_1} = 1$ and $\lambda_N$ such that $K_{\lambda_N} = m$, we initialize the values $\lambda_1$, $\lambda_N$ randomly, then run the convex clustering algorithm and check the resulting number of clusters. If $K_{\lambda_1} > 1$, we set $\lambda_1 = \alpha \lambda_1$ and similarly, if $K_{\lambda_N} < m$, we set $\lambda_N = \lambda_N / \alpha$. Once $\lambda_1$ and $\lambda_N$ such that $K_{\lambda_1} = 1$ and $K_{\lambda_N} = m$ are found, we choose $N$ equidistant points from the interval $[\lambda_N, \lambda_1]$. In our experiments, we used the initialization $\lambda_1 = \lambda_N = 0.1$, with $\alpha = 1.25$ and $N = 10$. The error metric is the same as in the preceding subsections. All results were averaged across 10 runs.

![Figure 3: Left: Performance of the proposed methods for linear regression, versus the number of samples available per user. We can see that the method that uses clusterpath matches the performance of the exact convex clustering method, for sufficiently large sample size. Right: Number of clusters produced by the two approaches to convex clustering for linear regression, versus the number of samples available per user. We can see that clusterpath matches the performance of exact convex clustering, for sufficiently large sample size.](image)

Figure 3 presents the performance of the two variants of the proposed method, using linear regression models. The left figure presents the MSE
versus the number of samples, while the right figure presents the number of clusters produced by convex clustering versus the number of samples. On $y$-axis in the left figure we plot the averaged normalized MSE, while on the $x$-axis we present the number of samples $n$ available to each user. We can again see that, for a small number of samples (less than 600), clusterpath performs worse than the exact convex clustering method, that chooses $\lambda$ based on (6). For larger values of $n$ ($n \geq 600$) we see that clusterpath consistently matches the performance of the exact convex clustering method, showing it achieves the order-optimal MSE rate. On $y$-axis in the right figure we plot the number of clusters produced by the convex clustering algorithm, while on $x$-axis we again plot the number of samples $n$. For smaller sample sizes, clusterpath produces less than the true $K = 4$ clusters, while the exact method produces roughly $m$ clusters, i.e., assigns each user to an individual cluster. These results can be explained by the fact that for small $n$, the interval (6) is likely to be empty, i.e., the upper bound that we take for our value of $\lambda$ smaller than the lower bound, thus creating many clusters. On the other hand, clusterpath always checks the recovery conditions based on the produced clustering, in step 2.(a). Convex clustering is known to have coarsening guarantees, i.e., merging multiple true clusters into one larger, e.g., [31], [33], in the form of intervals similar to (6). The coarsening intervals are larger than the corresponding intervals corresponding to the ground truth. Therefore, for smaller values of $n$, clusterpath is likely to produce a coarsening. As $n$ increases, we see that clusterpath is producing a larger number of clusters, eventually stabilizing on $K = 4$ clusters for $n \geq 600$. This is consistent with Figure 3 from the main body, as the clusterpath based method matches the performance of the exact method for $n \geq 600$.

### 5.4 Comparison with IFCA

In this subsection we compare the performance of IFCA and the proposed method (Subsection 3.1 in the main paper) on a synthetic dataset. We consider a linear regression problem, using the same setup as in Subsection 5.3, with $K = 4$.

Since we know the population optima, we can initialize the models of IFCA sufficiently close to the true population optima for each cluster. We initialize each model to be a random vector that is at least $\frac{1}{3}D$ close to the true population optima and at least $\frac{1}{5}D$ away from the true population optima, i.e., for $\theta^0_k \in \mathbb{R}^d$ the initialization of IFCA, we have $\frac{D}{5} \leq \|\theta^0_k - \theta^*_k\| \leq \frac{D}{3}$, for all $k \in [K]$. In our experiments, performing a different (random) initialization, we noted that the IFCA performance of IFCA significantly deteri-
orates if the initialization is not carefully tuned. In contrast, our method requires no specific initialization, hence offering another practical advantage over IFCA. Since the initialization is sufficiently close to the true population optima, we performed IFCA with model averaging (option 2 in their paper), where each user updates the received model for \( \tau > 1 \) local steps and the server averages the received models, corresponding to the same clusters. We chose 3 different step-sizes for IFCA, \( \alpha = 0.1, \alpha = 0.01 \) and \( \alpha = 0.05 \). For our method, we chose the value of \( \lambda \) as in Subsection 5.2. The results are averaged across 10 runs.

![Figure 4: Left: Performance of the proposed method and IFCA for \( n = 400 \) samples available to each user. We can see that IFCA matches the MSE of our method in a few rounds of communication, in the regime where our method is not order-optimal. Right: Performance of the proposed method and IFCA for \( n = 600 \) samples available to each user. We can see that, while IFCA converges relatively fast (about 100 communication rounds for highest precision), it is unable to achieve the same order-optimal performance in the regime where our method is order-optimal. On the other hand, our method achieves the order optimal-performance in a single communication round.](image)

Figure 4 presents the performance of our method and IFCA, using linear regression models. The left figure presents the performance when \( n = 400 \) samples is available to each user, while the right figure presents the performance when \( n = 600 \) samples is available to each user. On y-axis in both figures we present the normalized MSE, while on x-axis we present the number of communication rounds executed along the algorithm iterations. Here, by one communication round we mean the initial server-user broadcast and the returning user-server communication performed during one iteration of the algorithm (note that the order of communications is not important, e.g., our algorithm first has a user-server communication and then a server-user one, while each communication round of IFCA starts with a server-user communication and ends with a user-server communication). For our method,
the MSE is achieved in a single communication round and remains constant (clearly, our algorithm stops after one communication round, we just plot the horizontal line to facilitate the comparison). We can see on the left figure that, for $n = 400$, IFCA requires around 20 communication rounds to match the performance of our method, for step-sizes 0.1 and 0.05, with the performance not improving significantly with each additional communication round. For the step-size 0.01, IFCA requires around 70 communication rounds to match the performance of our method, while improving the overall performance by an order of magnitude after approximately 120 communication rounds. We note that $n = 400$ represents the phase-transitional regime, i.e., a regime in which our method has moved away from clustering each user to a single cluster, but is not yet order-optimal. On the other hand, from the right figure we can see that, for $n = 600$, even after 1000 communication rounds, IFCA is unable to match the performance of our method, achieved in a single communication round. We note that $n = 600$ is the order-optimal regime for our method, i.e., it is above the threshold shown in Theorem 1, therefore representing the regime in which all the communication beyond the first round is redundant, as verified by the right figure.

6 Conclusion

We proposed a one-shot approach for CFL, based on a simple inference and averaging scheme. The proposed approach is communication efficient, as it requires a single round of communication. Moreover, our theoretical analysis showed that the method provides order-optimal MSE rates, in terms of the sample size. Compared to the state-of-the-art algorithms that require multiple rounds of communication, the proposed method improves existing results by a factor that is logarithmic in the total number of samples in the system and provides significant communication reduction. Unlike other existing methods, the proposed method does not require knowledge of the underlying number of clusters $K$. Numerical experiments corroborate our findings.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, ser. Proceedings
[2] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” arXiv preprint arXiv:1610.05492, 2016.

[3] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, “Qsgd: Communication-efficient sgd via gradient quantization and encoding,” in Advances in Neural Information Processing Systems, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/6c340f25839e6acdc73414517203f5f0-Paper.pdf

[4] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, “Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization,” in Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. PMLR, 2020, pp. 2021–2031. [Online]. Available: https://proceedings.mlr.press/v108/reisizadeh20a.html

[5] A. Koloskova, S. Stich, and M. Jaggi, “Decentralized stochastic optimization and gossip algorithms with compressed communication,” in Proceedings of the 36th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 3478–3487. [Online]. Available: https://proceedings.mlr.press/v97/koloskova19a.html

[6] S. Stich, “Local sgd converges fast and communicates little,” in ICLR 2019-International Conference on Learning Representations, no. CONF, 2019.

[7] A. Khaled, K. Mishchenko, and P. Richtarik, “Tighter theory for local sgd on identical and heterogeneous data,” in Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. PMLR, 2020, pp. 4519–4529. [Online]. Available: https://proceedings.mlr.press/v108/bayoumi20a.html
[8] K. Mishchenko, G. Malinovsky, S. Stich, and P. Richtarik, “ProxSkip: Yes! Local gradient steps provably lead to communication acceleration! Finally!” in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 2022, pp. 15750–15769. [Online]. Available: https://proceedings.mlr.press/v162/mishchenko22b.html

[9] Y. Zhang, J. C. Duchi, and M. J. Wainwright, “Communication-efficient algorithms for statistical optimization,” *Journal of Machine Learning Research*, vol. 14, no. 104, pp. 3321–3363, 2013. [Online]. Available: http://jmlr.org/papers/v14/zhang13b.html

[10] N. Guha, A. Talwalkar, and V. Smith, “One-shot federated learning,” *arXiv preprint arXiv:1902.11175*, 2019.

[11] Y. Zhou, G. Pu, X. Ma, X. Li, and D. Wu, “Distilled one-shot federated learning,” *arXiv preprint arXiv:2009.07999*, 2020.

[12] S. Salehkaleybar, A. Sharifi Nassab, and S. J. Golestani, “One-shot federated learning: Theoretical limits and algorithms to achieve them,” *Journal of Machine Learning Research*, vol. 22, no. 189, pp. 1–47, 2021. [Online]. Available: http://jmlr.org/papers/v22/19-1048.html

[13] D. K. Dennis, T. Li, and V. Smith, “Heterogeneity for the win: One-shot federated clustering,” in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 2021, pp. 2611–2620. [Online]. Available: https://proceedings.mlr.press/v139/dennis21a.html

[14] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, “Tackling the objective inconsistency problem in heterogeneous federated optimization,” *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

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

[16] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, “Federated multi-task learning,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates,
[17] F. Hanzely and P. Richtárik, “Federated learning of a mixture of global and local models,” arXiv preprint arXiv:2002.05516, 2020.

[18] F. Hanzely, S. Hanzely, S. Horváth, and P. Richtárik, “Lower bounds and optimal algorithms for personalized federated learning,” Advances in Neural Information Processing Systems, vol. 33, pp. 2304–2315, 2020.

[19] A. Fallah, A. Mokhtari, and A. Ozdaglar, “Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach,” in Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 3557–3568. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/24389bfe4fe2e6ba8b9aa9203a44cdad-Paper.pdf

[20] M. Khodak, M.-F. F. Balcan, and A. S. Talwalkar, “Adaptive gradient-based meta-learning methods,” in Advances in Neural Information Processing Systems, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/f4aa0dd960521e045ae2f20621040e9-Paper.pdf

[21] C. T Dinh, N. Tran, and J. Nguyen, “Personalized federated learning with moreau envelopes,” Advances in Neural Information Processing Systems, vol. 33, pp. 21394–21405, 2020.

[22] T. Li, S. Hu, A. Beirami, and V. Smith, “Ditto: Fair and robust federated learning through personalization,” in International Conference on Machine Learning. PMLR, 2021, pp. 6357–6368.

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

[24] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, “An efficient framework for clustered federated learning,” IEEE Transactions on Information Theory, pp. 1–1, 2022.

[25] A. Ghosh, J. Hong, D. Yin, and K. Ramchandran, “Robust federated learning in a heterogeneous environment,” arXiv preprint arXiv:1906.06629, 2019.
[26] F. Sattler, K.-R. Müller, and W. Samek, “Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints,” IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 8, pp. 3710–3722, 2021.

[27] A. Armacki, D. Bajovic, D. Jakovetic, and S. Kar, “Personalized federated learning via convex clustering,” arXiv preprint arXiv:2202.00718, 2022.

[28] K. Sridharan, S. Shalev-Shwartz, and N. Srebro, “Fast rates for regularized objectives,” in Advances in Neural Information Processing Systems, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., vol. 21. Curran Associates, Inc., 2008. [Online]. Available: https://proceedings.neurips.cc/paper/2008/file/abdd815286ba1007abfbb8415b83ae2cf-Paper.pdf

[29] A. W. v. d. Vaart, Asymptotic Statistics, ser. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998.

[30] L. Zhang, T. Yang, and R. Jin, “Empirical risk minimization for stochastic convex optimization: $O(1/n)$- and $O(1/n^2)$-type of risk bounds,” in Proceedings of the 2017 Conference on Learning Theory, ser. Proceedings of Machine Learning Research, S. Kale and O. Shamir, Eds., vol. 65. PMLR, 2017, pp. 1954–1979. [Online]. Available: https://proceedings.mlr.press/v65/zhang17a.html

[31] A. Panahi, D. Dubhashi, F. D. Johansson, and C. Bhattacharyya, “Clustering by sum of norms: Stochastic incremental algorithm, convergence and cluster recovery,” in Proceedings of the 34th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 2017, pp. 2769–2777. [Online]. Available: https://proceedings.mlr.press/v70/panahi17a.html

[32] T. D. Hocking, A. Joulin, F. Bach, and J.-P. Vert, “Clusterpath: An algorithm for clustering using convex fusion penalties,” in Proceedings of the 28th International Conference on International Conference on Machine Learning, ser. ICML’11. Madison, WI, USA: Omnipress, 2011, p. 745–752.

[33] D. Sun, K.-C. Toh, and Y. Yuan, “Convex clustering: Model, theoretical guarantee and efficient algorithm,” Journal of Machine
Learning Research, vol. 22, no. 9, pp. 1–32, 2021. [Online]. Available: http://jmlr.org/papers/v22/18-694.html

[34] E. C. Chi and K. Lange, “Splitting methods for convex clustering,” Journal of Computational and Graphical Statistics, vol. 24, no. 4, pp. 994–1013, 2015, pMID: 27087770. [Online]. Available: https://doi.org/10.1080/10618600.2014.948181

[35] S. Lloyd, “Least squares quantization in pcm,” IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 129–137, 1982.

[36] A. Banerjee, S. Merugu, I. S. Dhillon, J. Ghosh, and J. Lafferty, “Clustering with bregman divergences.” Journal of machine learning research, vol. 6, no. 10, 2005.

[37] A. Armacki, D. Bajovic, D. Jakovetic, and S. Kar, “Gradient based clustering,” in Proceedings of the 39th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 2022, pp. 929–947. [Online]. Available: https://proceedings.mlr.press/v162/armacki22a.html

[38] W. Ben-Ameur, P. Bianchi, and J. Jakubowicz, “Robust distributed consensus using total variation,” IEEE Transactions on Automatic Control, vol. 61, no. 6, pp. 1550–1564, 2016.

[39] H. Robbins and S. Monro, “A Stochastic Approximation Method,” The Annals of Mathematical Statistics, vol. 22, no. 3, pp. 400 – 407, 1951. [Online]. Available: https://doi.org/10.1214/aoms/1177729586

[40] A. Rakhlin, O. Shamir, and K. Sridharan, “Making gradient descent optimal for strongly convex stochastic optimization,” in Proceedings of the 29th International Conference on International Conference on Machine Learning, 2012, pp. 1571–1578.

[41] A. Defazio, F. Bach, and S. Lacoste-Julien, “Saga: A fast incremental gradient method with support for non-strongly convex composite objectives,” in Advances in Neural Information Processing Systems, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/ede7e2b6d13a41ddl9f4bde84fde737-Paper.pdf
R. Johnson and T. Zhang, “Accelerating stochastic gradient descent using predictive variance reduction,” in Advances in Neural Information Processing Systems, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013. [Online]. Available: https://proceedings.neurips.cc/paper/2013/file/ac1dd209cbbc5e5d1c6e28598e8ebbe8-Paper.pdf

N. J. Harvey, C. Liaw, and S. Randhawa, “Simple and optimal high-probability bounds for strongly-convex stochastic gradient descent,” arXiv preprint arXiv:1909.00843, 2019.

S. Diamond and S. Boyd, “CVXPY: A Python-embedded modeling language for convex optimization,” Journal of Machine Learning Research, vol. 17, no. 83, pp. 1–5, 2016.

A Appendix

In this Appendix we show conditions under which it is suitable for two users with different distributions to form a cluster by merging their respective data. As in the main body, we use $\theta^*_k$ and $\hat{\theta}_i$ to denote the population optimal of cluster $k \in [K]$ and local ERM of user $i \in [m]$, respectively, i.e., $\theta^*_k = \arg\min_{\theta \in \Theta} F_k(\theta)$ and $\hat{\theta}_i = \arg\min_{\theta \in \Theta} f_i(\theta)$. We then have the following result.

Lemma 7. Let Assumptions 2 and 3 hold and additionally assume $\ell$ is strongly convex, with strong convexity parameter $\mu$. Let all users sample data from a unique distribution, i.e., each user $i$ samples data following $D_i$, $i \in [m]$. Denote by $D_{i+j}$ the mixture of distributions $D_i$ and $D_j$, i.e., the distribution such that $F_{i+j}(\theta) = p_i F_i(\theta) + p_j F_j(\theta)$, where $0 < p_i, p_j < 1$, such that $p_i + p_j = 1$. If the distributions $D_i$ and $D_j$ are such that $\|\theta^*_i - \theta^*_j\|^2 \leq \epsilon$,

then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$

$$\|\hat{\theta}_{i+j} - \theta^*_m\|^2 = \mathcal{O}\left(\frac{\log \left(\frac{1}{\delta}\right)}{n_i + n_j} + \epsilon\right),$$

with $m \in \{i, j\}$ and $\hat{\theta}_{i+j} = \arg\min_{\theta \in \Theta} p_i f_i(\theta) + p_j f_j(\theta)$.

Some remarks are now in order.
Remark 22. Lemma 7 roughly tells us that, as long as \( \frac{1}{n_i + n_j} + \epsilon < \min \left\{ \frac{1}{n_i}, \frac{1}{n_j} \right\} \), i.e., \( \epsilon < \frac{\min\{n_i, n_j\}}{\max\{n_i, n_j\}(n_i + n_j)} \), we have that the model trained on the joint datasets of users \( i \) and \( j \) is beneficial to both users. For example, when \( n_i = n \), \( \forall i \in [m] \), the condition on \( \epsilon \) evaluates to \( \epsilon < \frac{1}{2n} \).

Remark 23. If \( \frac{1}{n_i + n_j} + \epsilon < \min \left\{ \frac{1}{n_i}, \frac{1}{n_j} \right\} \), Lemma 7 tells us that it is beneficial to treat the users \( i \) and \( j \) as belonging to the same cluster. Therefore, averaging the local ERM \s trained by users \( i \) and \( j \) leads to mutual benefits, even though the two users come from different, but mutually close distributions (as measured by the distance of the population optimia). Therefore, by treating users \( i \) and \( j \) as coming from the same distribution \( D_k \), where \( k = i + j \), the number of underlying distributions (and hence clusters) is reduced to some \( K < m \). On the other hand, by the mere presence of heterogeneity, i.e., non-IID data across users, we have \( K > 1 \), hence \( 1 < K < m \) in Assumption 1 is justified.

Proof of Lemma 7. Applying the results of [28], we have that, with probability at least \( 1 - \delta \)

\[
\| \hat{\theta}_i - \theta_i^* \|^2 = O \left( \frac{\log \left( \frac{1}{\delta} \right)}{n_i} \right).
\]

Let \( D_{i+j} \) denote the mixture distribution of distributions \( D_i \) and \( D_j \). For ease of notation, let \( k := i + j \). Denote by \( \theta_k^* \) the population optima of the mixture distribution \( D_k \). We then have

\[
\| \hat{\theta}_k - \theta_k^* \|^2 \leq 2\| \hat{\theta}_k - \theta_k^* \|^2 + 2\| \theta_k^* - \theta_i^* \|^2 \leq O \left( \frac{\log \left( \frac{1}{\delta} \right)}{n_i + n_j} \right) + 2\| \theta_k^* - \theta_i^* \|^2, \tag{28}
\]

where the second inequality again follows from [28]. Using strong convexity of \( F \)'s, we have that

\[
\frac{\mu}{2} \| \theta_k^* - \theta_i^* \|^2 \leq F_k(\theta_i^*) - F_k(\theta_k^*) = p_i F_i(\theta_i^*) + p_j F_j(\theta_i^*) - p_i F_i(\theta_k^*) - p_j F_j(\theta_k^*) \leq p_j (F_j(\theta_i^*) - F_j(\theta_k^*)) = p_j (F_j(\theta_i^*) - F_j(\theta_j^*) + F_j(\theta_j^*) - F_j(\theta_k^*)) \leq p_j (F_j(\theta_i^*) - F_j(\theta_j^*)),
\]

48
where we used the fact that $\theta^*_m = \arg\min_{\theta \in \Theta} F_m(\theta)$, $m \in \{i, j\}$, in the second and third inequalities, respectively. Finally, using $L$-Lipschitz continuous gradients of $F$’s, we get that

$$p_j \left( F_j(\theta^*_i) - F_j(\theta^*_j) \right) \leq \frac{p_j L}{2} \|\theta^*_i - \theta^*_j\|^2 = \frac{p_j L}{2} \epsilon. \tag{29}$$

Combining (28) and (29), we get that, with probability at least $1 - \delta$

$$\|\hat{\theta}_k - \theta^*_i\|^2 = O \left( \frac{\log \left( \frac{1}{\delta} \right)}{n_i + n_j} \right) + \frac{2p_j L}{\mu} \epsilon. \tag{30}$$

Repeating the same steps for user $j$, we get that, for $m \in \{i, j\}$, with probability at least $1 - \delta$

$$\|\hat{\theta}_k - \theta^*_m\|^2 = O \left( \frac{\log \left( \frac{1}{\delta} \right)}{n_i + n_j} \right) + \frac{2p L}{\mu} \epsilon,$$

where $p = \max\{p_i, p_j\}$. $\square$