Fed-SC: One-Shot Federated Subspace Clustering over High-Dimensional Data

Songjie Xie*, Youlong Wu*, Kewen Liao†, Lu Chen‡, Chengfei Liu‡, Haifeng Shen†, MingJian Tang§, Lu Sun*
*School of Information Science and Technology, ShanghaiTech University, Shanghai, China
†HilstLab, Peter Faber Business School, Australian Catholic University, Sydney, Australia
‡Department of Computing Technologies, Swinburne University of Technology, Melbourne, Australia §Atlassian
{xiesj, wuyl1, sunlu1}@shanghaitech.edu.cn, {kewen.liao, haifeng.shen}@acu.edu.au,
{luchen, cliu}@swin.edu.au, mtang2@atlassian.com

***Abstract*—Recent work has explored federated clustering and developed an efficient k-means based method. However, it is well known that k-means clustering underperforms in high-dimensional space due to the so-called "curse of dimensionality". In addition, high-dimensional data (e.g., generated from health-care, medical, and biological sectors) are pervasive in the big data era, which poses critical challenges to federated clustering in terms of, but not limited to, clustering effectiveness and communication efficiency. To fill this significant gap in federated clustering, we propose a one-shot federated subspace clustering scheme Fed-SC that can achieve remarkable clustering effectiveness on high-dimensional data while keeping communication cost low using only one round of communication for each local device. We further establish theoretical guarantees on the clustering effectiveness of one-shot Fed-SC and exploit the benefits of statistical heterogeneity across distributed data. Extensive experiments on synthetic and real-world datasets demonstrate significant effectiveness gains of Fed-SC compared with both subspace clustering and one-shot federated clustering methods.**

***Index Terms*—Federated clustering, subspace clustering, high-dimensional data, statistical heterogeneity**

I. Introduction

Federated clustering (FC) [1]–[4] aims to cluster the data distributed over a heterogeneous device network comprised of mobile phones, wearables, etc., known as a *federated network* [5]. An FC method generally works in a restricted client/server model, where raw data on client devices are not allowed to be disclosed. Specifically, each client device first clusters its local data and then sends privacy-preserved clustering information to the central server. The server aggregates the local clustering information to generate global clusters and then communicates the global clustering information back to the clients for their local cluster updates. The above process may repeat till a satisfactory clustering result is obtained.

FC can serve as a data clustering regime for cases where distributed data cannot be exposed or gathered due to legal or privacy reasons, e.g., clustering healthcare, medical, or genomics data residing at different centers. In these cases, sensitive data are kept locally as exchanging information such as raw data among clients and the server poses a serious privacy leakage threat. Recently, the method of one-shot FC, called k-FED, was proposed in [1], which is based on k-means and considers using only one round of communication between the server and devices, so as to speed up the clustering process and provide better privacy-preserving. k-FED was shown to perform very well on low-dimensional data assuming the number of global clusters is significantly larger than the number of local clusters at each device, i.e., a notion of statistical heterogeneity.

Corresponding authors: Youlong Wu and Kewen Liao

Nevertheless, high-dimensional data such as healthcare records [6], medical images [7], and sequencing data [8] can render k-FED much less effective as k-means-based clustering methods commonly suffer from the curse of dimensionality. To cluster high-dimensional data distributed over a federated network, we may first apply dimension-reduction techniques to the local client data and then perform k-FED. However, this makes the clustering effectiveness guarantee of [1] invalid and leads to arbitrary clustering results even much worst than k-FED (see Table IV in the later experiments section). Therefore, for the high-dimensional data distributed over a federated network, it remains an open challenge to achieve high clustering effectiveness.

Furthermore, in order to develop an effective FC approach that is also communication-efficient, we choose to extend centralized subspace clustering (SC) methods [9], [10] to federated networks. SC emerges as a promising approach for high-dimensional data clustering, as in many applications high-dimensional data can be well represented by a union of low-dimensional subspaces [11]. Compared to the conventional high-dimensional data analysis approaches [12], SC demonstrates strong empirical performance with theoretical guarantee [13] and has been successfully applied in many areas like computer vision [14], image processing [15], and bioinformatics [16]. In order to ensure communication efficiency, we aim to develop a one-shot method for federated SC. However, bearing data privacy[1] in mind, in a federated network the central server is prohibited from knowing the raw data, which often leads to inferior solution effectiveness unless multiple rounds of communication are allowed between clients and the server [5]. Therefore, it is non-trivial to devise

[1]While privacy is not the focus of this work, the proposed method has been designed to reduce communication overhead and potential information leakage via sampling. Specifically, the method only requires a single communication round that relies on a limited number of randomly generated samples. More privacy aspect of the method is discussed in Remark 2.

a federated SC scheme that meets one-shot communication and high clustering effectiveness concurrently.

In this work, we propose such a scheme called Fed-SC, where each client device samples its local subspaces in a uniform way and sends the sampled results to the central server instead of uploading the entire subspaces or data (see Fig. 2 in Section IV). To verify the clustering effectiveness of Fed-SC, we establish theoretical guarantees following the works of centralized SC [9], [10], [17] and empirically demonstrate its performance through extensive experiments on both synthetic and real-world datasets. Our main contributions can be summarized as follows:

- We propose a one-shot federated SC scheme named Fed-SC. To the best of our knowledge, this is the first work introducing SC to the FC regime, which enables FC to deal with high-dimensional data.
- We provide theoretical guarantees on the effectiveness of Fed-SC using the criteria of self-expressiveness property and exact clustering. Our theoretical analysis formulates the conditions when the proposed Fed-SC reaches the optimum and provides insights on the subspace property and data distribution that can lead to a good clustering performance.
- We conduct extensive experiments on both synthetic and real-world datasets to demonstrate the practical advantages of Fed-SC, compared to state-of-the-art centralized SC and one-shot FC methods.

The rest of the paper is organized as follows. Section II introduces the related work. Section III presents preliminaries and formulates the problem. Section IV describes the proposed federated SC scheme. The theoretical guarantees on the clustering effectiveness are formally provided in Section V. Experimental results on synthetic and real-world datasets are reported in Section VI, followed by concluding remarks in Section VII.

II. Related Work

This section briefly reviews the centralized SC methods and the works of distributed and federated clustering, alongside with discussions of the issues in designing a federated SC scheme that motivate our research.

A. Centralized Subspace Clustering

SC is a common approach to finding the informative clusters of high-dimensional datasets [11], [18]. Many popular SC algorithms [19]–[23] first construct an affinity graph of data points according to the subspaces they lie, and then apply spectral clustering [24] to segment data. Among the algorithms, sparse subspace clustering (SSC) [9] emerged as a popular method due to its theoretical guarantee and empirical success. To alleviate the prohibitive computational complexity of SSC, its version with orthogonal matching pursuit (SSCOMP) was proposed in [25] to improve the scalability. Besides, an oracle-based active set algorithm (EnSC) [26] was introduced to solve the SSC optimization problems with additional regularization. Apart from the SSC-like algorithms, there exists another line of work developed to construct the affinity graph by calculating the similarity between data points. Thresholding-based subspace clustering (TSC) [10] was developed by thresholding the cosine distances between data points, and establishing theoretical guarantees that rely critically on the uniform distribution of data points on subspaces. A simple and efficient greedy algorithm NSN [27] was proposed to estimate the underlying subspaces and cluster the high-dimensional data. However, existing SC methods were developed for centralized settings and are not well-suited for deployment in federated networks. These methods exhibit prohibitive computational complexity, and their straightforward adoption to FL would result in the exposure of the entire underlying subspaces during computation, thereby violating the stringent communication constraints and privacy requirements that are inherent to federated networks.

B. Distributed and Federated Clustering

Distributed clustering (DC). Many distributed implementations of centralized clustering algorithms were proposed to handle large-scale datasets in distributed settings. DC is different from FC inherently. DC methods require a large amount of information transmission and expose the entire underlying data structure over the distributed networks, which FC forbids. For the classical clustering methods such as k-means and DBSCAN [28], their distributed implementations [29]–[32] exploit the resource of multiprocessors for the computation of distances among the data points. For distributed SC methods, the algorithms [33]–[35] were proposed to distributedly compute the representation matrix.

Federated clustering (FC). In supervised federated learning (FL), clustering has been widely used to address heterogeneity issues (e.g., statistical heterogeneity [36] and system heterogeneity [37]) by clustering client devices. On the other hand, clustering the data residing on federated networks is relatively unexplored. To deploy clustering algorithms in federated networks, there are certain attempts to explore unsupervised FL. Federated architecture proposed in [3] conducts unsupervised representation learning to pretrain deep neural networks using unlabeled data. Other authors [4] focused on federated representation learning and introduced a federated algorithm with a dictionary and alignment. Recently, the work of [1] proposed a one-shot federated k-means algorithm and rigorously proved the benefits of statistical heterogeneity on federated k-means. In [2], a federated unsupervised clustering method was developed by training neural generative models across the devices. However, the method requires multiple rounds of communication and faces the challenge of high communication cost. While the federated methods based on k-means and generative models reveal the unique advantages of FC, it remains unexplored to design an effective and efficient federated scheme for clustering pervasive high-dimensional data. This motivates us to investigate SC in a federated setting and design a one-shot scheme with theoretical guarantees.

TABLE I
SUMMARY OF NOTATIONS

Notation	Meaning
Z, z	The number of devices and the index of the device.
L, ℓ	The number of subspaces and the index of subspace.
$\mathbf{X}_\ell, \mathbf{X}^{(z)}$	Submatrix where columns are distributed on subspace $\mathcal{S}_\ell$ and the data matrix on local device z.
$\mathbf{X}_\ell^{(z)}$	Matrix of data points that distributed on $\mathcal{S}_\ell$ in local device z.
$L^{(z)}$	Number of subspaces where the local data $\mathbf{X}^{(z)}$ is distributed.
Z_ℓ	Number of devices that contains the data from subspace $\mathcal{S}_\ell$.
$T^{(z)}, r^{(z)}$	Data partitions and the number of partitions on device z.
$\mathbf{\Theta}^{(z)}$	Samples generated on device z, $\mathbf{\Theta}^{(z)} = [\theta_t^{(z)}]_{t=1}^{r^{(z)}}$.
$\tau_i^{(z)}$	Cluster assignment of the generated sample $\theta_i^{(z)}$ with $\tau_i^{(z)} \in [L]$.
$\hat{T}^{(z)}$	Updated partitions of local data on device z.
$\mathbf{X}_{\ell,-j}$	$[\mathbf{x}_{\ell,1}, \ldots, \mathbf{x}_{\ell,j-1}, \mathbf{x}_{\ell,j+1}, \ldots, \mathbf{x}_{\ell,N_\ell}]$
$\mathbb{P}_\ell$	The projection matrix to subspace $\mathcal{S}_\ell$.
$\mathcal{P}(\mathbf{X})$	The symmetrized convex hull of the columns of $\mathbf{X}$, i.e., conv$(\pm\mathbf{x}_1, \ldots, \pm\mathbf{x}_N)$.
$\\|\mathbf{x}\\|_p$	The ℓ_p-norm of a vector $\mathbf{x} \in \mathbb{R}^n$, defined as $\\|\mathbf{x}\\|_p \triangleq (\sum_{j=1}^n \|x_j\|^p)^{\frac{1}{p}}$, where $\|\cdot\|$ denotes the absolute value.

III. Preliminaries and Problem Formulation

In this section, we first introduce the problem of subspace clustering including the existing methods. Then, we cover preliminaries that characterize the subspace clustering problem in federated networks. The notations frequently used in this paper are summarized in Table I.

A. Subspace Clustering

Subspace clustering (SC). There are N high-dimensional data points in $\mathbb{R}^n$, denoted by $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{n \times N}$. The data points are lying on a union of L unknown subspaces $\bigcup_{\ell=1}^L \mathcal{S}_\ell$ of unknown dimensions $\{d_\ell\}_{\ell=1}^L$. Subspace clustering (SC) aims to cluster the data points $\mathbf{X}$ into L groups according to the subspaces in which they lie.

Spectral-based SC method. The spectral-based SC methods uncover the low-dimensional unknown subspaces $[\mathcal{S}_\ell]_{\ell=1}^L$ from data $\mathbf{X}$ by two major steps: 1) constructing an affinity graph $\mathbf{W} \in \mathbb{R}^{N \times N}$ of N data points where the affinity characterizes whether two points lie in the same subspace, and 2) applying spectral clustering [24] on $\mathbf{W}$ to generate L clusters. Obviously, the first step is the most critical in SC since the success of spectral clustering depends on the construction of $\mathbf{W}$. There are two popular methods for constructing $\mathbf{W}$, which are introduced as follows.

Sparse vector based $\mathbf{W}$. Sparse subspace clustering (SSC) constructs $\mathbf{W}$ by solving the following optimization problem such that each data point $\mathbf{x}_i \in \mathbf{X}$ is expressed as a linear combination of other data points:

$$\min_{\mathbf{c}_i \in \mathbb{R}^N} \|\mathbf{c}_i\|_1, \quad \text{s.t.} \quad \mathbf{x}_i = \mathbf{X}\mathbf{c}_i, c_{ii} = 0, \tag{1}$$

where $\mathbf{c}_i \in \mathbb{R}^N$ is the sparse solution for $\mathbf{x}_i$. By arranging the solutions into a matrix $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N] \in \mathbb{R}^{N \times N}$, the affinity graph $\mathbf{W}$ is constructed by $\mathbf{W} = |\mathbf{C}| + |\mathbf{C}|^T$. To handle noisy datasets, the Lasso version of SSC is a practical extension often used in numerous applications:

$$\min_{\mathbf{c}_i \in \mathbb{R}^N} \frac{\lambda}{2}\|\mathbf{X}\mathbf{c}_i - \mathbf{x}_i\|_2^2 + \|\mathbf{c}_i\|_1, \quad \text{s.t.} \quad c_{ii} = 0. \tag{2}$$

Thresholding-based $\mathbf{W}$. Another line of work, thresholding-based subspace clustering (TSC) [10], constructs $\mathbf{W}$ by calculating and thresholding the cosine distances between data points. In $\mathbf{W}$, each data point is only connected to its q nearest neighbors in spherical distance, where q is a hyperparameter. However, compared to SSC, the effectiveness of TSC heavily relies on uniform distribution of data points and it does not work well on some real datasets (see experiments in [10]).

Criteria of effective clustering. There are two main criteria for evaluating SC results, according to [9], [17].

Self-expressiveness property (SEP). SEP ensures that no two points from different subspaces are connected in the constructed affinity graph $\mathbf{W}$. In other words, SEP guarantees no false connections in $\mathbf{W}$, so spectral clustering detects correct subspaces. However, SEP allows data points belonging to the same subspace to form multiple connected components. This can cause spectral clustering to over-segment the data.

Exact clustering. This stronger criterion ensures data points from the same subspace form a single connected component in the constructed affinity graph $\mathbf{W}$. As such, the spectral clustering step can segment the data with no error [24].

B. Federated Subspace Clustering

We now consider SC in a federated network with Z client devices, where each client device stores a subset of $\mathbf{X}$ and can communicate with a central server. Formally, we index local devices by $z \in [Z]$ and denote the local data matrix contained in device z by $\mathbf{X}^{(z)} \in \mathbb{R}^{n \times N^{(z)}}$, where $N^{(z)}$ is number of data points in device z with $\sum_{z=1}^Z N^{(z)} = N$. Let $\mathbf{X}_\ell^{(z)} \in \mathbb{R}^{n \times N_\ell^{(z)}}$ denotes $N_\ell^{(z)}$ local data points distributed on $\mathcal{S}_\ell$ in device z. Note that some $\mathbf{X}_\ell^{(z)}$ could be empty, we use $L^{(z)}$ to denote the number of non-empty data matrix on local device z with $1 \leq L^{(z)} \leq L$. Not all the devices contain data points distributed on some subspaces if $L^{(z)} < L$. Therefore, we use Z_ℓ to denote the number of devices containing the data from subspace $\mathcal{S}_\ell$ with $1 \leq Z_\ell \leq Z$. An illustration of such a federated setting for SC is shown in Fig. 1.

Research problem. Given high-dimensional data $\mathbf{X}$ residing in a federated network with Z devices, federated SC aims to cluster $\mathbf{X}$ into L classes according to the global subspaces $\{\mathcal{S}_\ell\}_{\ell=1}^L$ they lie. The developed federated SC method is expected to minimize communication cost and local information exposure. Meanwhile, its effectiveness should be theoretically guaranteed in terms of SEP and exact clustering. We are now ready to introduce our proposed federated SC method in the next section.

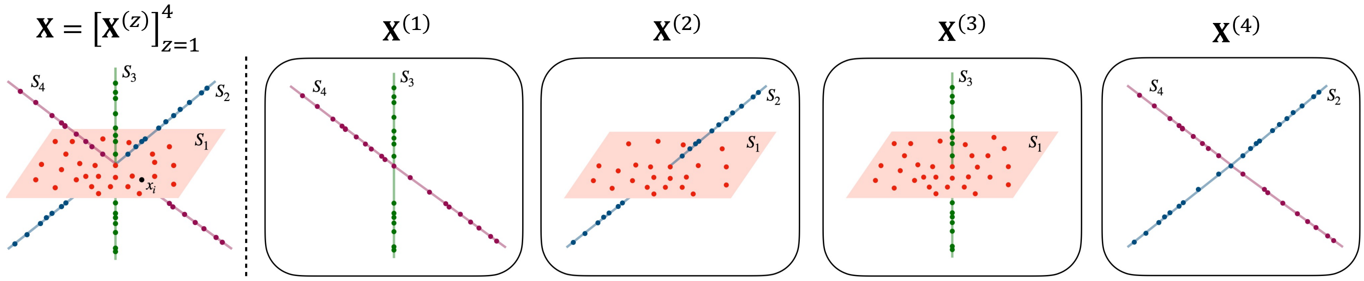

Fig. 1. The data points $\mathbf{X}$ are distributed on the union of four subspaces $\bigcup_{\ell=1}^{4} \mathcal{S}_\ell \in \mathbb{R}^3$. There are four local devices containing the subset of data points and statistical heterogeneity exists because each local data $\mathbf{X}^{(z)}$ is distributed on a union of only two subspaces. In this case, $L^{(z)} = 2$ for each device $z \in [4]$ and $Z_\ell = 2$ for each subspace $\mathcal{S}_\ell, \ell \in [4]$.

Algorithm 1 Fed-SC scheme

Input: The total data matrix $\mathbf{X} = [\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \dots, \mathbf{X}^{(Z)}]$ and the number of subspaces L

// Phase 1: local clustering and sampling
for $z \leftarrow 1$ **to** Z **do**
  Client z runs Algorithm 2 with $\mathbf{X}^{(z)}$ and sends the generated samples $\mathbf{\Theta}^{(z)}$ to the central server;
end for
// Phase 2: central clustering
Server runs SC algorithm (TSC or SSC) to segment $[\mathbf{\Theta}^{(z)}]_{z=1}^{Z}$ into L clusters;
for $z \leftarrow 1$ **to** Z **do**
  Server delivers $\{\tau_t^{(z)}\}_{t=1}^{r^{(z)}}$ to client z;
end for
// Phase 3: Local update
Each client z updates $T^{(z)}$ into $\hat{T}^{(z)} = (\hat{T}_\ell^{(z)})_{\ell=1}^{L}$ by
$$\hat{T}_\ell^{(z)} = \{i : i \in T_t^{(z)} \text{ and } \tau_t^{(z)} = \ell\} \text{ for each } \ell \in [L];$$

Output: The partitions $(\hat{T}^{(1)}, \hat{T}^{(2)}, \dots, \hat{T}^{(Z)})$ of the total data $\mathbf{X}$

IV. Federated Subspace Clustering

In this section, we propose a federated scheme for SC on high-dimensional data, denoted by Fed-SC, and discuss its practical advantages.

A. Federated Scheme

The overall federated scheme of SC is shown in Algorithm 1 and Fig 2. We first apply SC on each local device, which leads to initial local clusters resident in each device. To securely send the clustering information to the central server while preserving the clustering effectiveness, we propose to *encode and then sample local clusters* on each device, and the server just collects samples of the encoded clustering information. After that, the server performs SC on the collected samples and generates a global clustering result. Finally, the global result is sent back to each local device so that each local data point can be assigned to a global cluster.

The above one-shot process looks straightforward. However, in order to provide guarantees on the clustering effectiveness, several challenges as below need to be addressed, which do not exist in the centralized SC.

Handling heterogeneous data[2]. Local data residing on different devices could have very different characteristics. The setting of clustering parameters required by SC on each device should be dependent on the local data distribution. Therefore, we only choose to run SSC for local clustering instead of TSC which requires a uniformness assumption and a thresholding parameter q. Also for SSC deployed in a federated network, the number of local subspaces $L^{(z)}$ (as the clustering parameter) is unknown and non-uniform across devices. These need to be estimated for each z from local data distribution, unlike the centralized case where L is usually given.

Ensuring clustering effectiveness. Making Fed-SC as effective as centralized FC is challenging due to a lack of complete data information at the central server. We propose an effective encoding and uniform sampling method that can nicely capture the local cluster information to guarantee the overall effectiveness of Fed-SC.

In the following subsections, we discuss each step of Fed-SC in great detail, which addresses the above challenges. We leave theoretical guarantees and the corresponding rigorous proofs to the next section.

B. Local Clustering

In this part, we focus on discussing how a device z clusters local data, and the following presented method applies to every $z \in [Z]$. Note that SSC can generate $\mathbf{W}$ according to local data characteristics instead of manually thresholding for z like TSC. We propose to employ SSC on device z to better handle heterogeneous distributed data.

We shall construct sparse vector based $\mathbf{W}$. Each local device solves the SSC optimization problem (1) in case of noiseless data or (2) for noisy data, to obtain the self-expression matrix $\mathbf{C}^{(z)}$ for local data points $\mathbf{X}^{(z)}$. The affinity graph $\mathbf{W}^{(z)}$ is formed by $|\mathbf{C}^{(z)}| + |\mathbf{C}^{(z)}|^T$.

In order to apply spectral clustering on $\mathbf{W}^{(z)}$, a parameter, the number of clusters, must be provided. In the centralized SC, this is usually provided by default. However, in the federated network with Z client devices, local data on each

[2]Throughout the paper, a federated network is said with statistical heterogeneity if there exists at least one device z such that $L^{(z)} < L$. Also, the smaller $L^{(z)}$'s, the more significant statistical heterogeneity exists across the federated network.

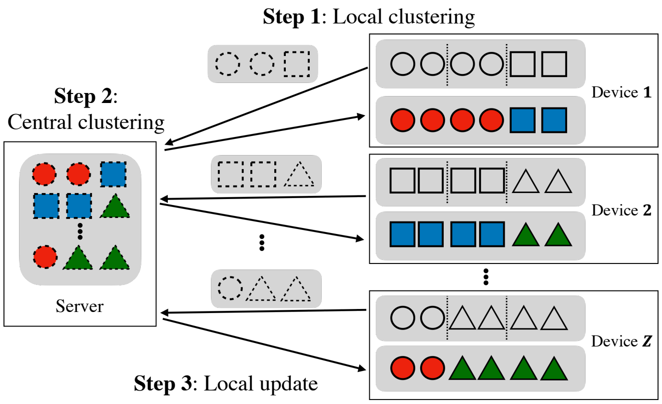

Fig. 2. Main steps of the proposed Fed-SC. Firstly, each client device conducts a local SC to segment its data (i.e., shapes without color) into different partitions according to their underlying subspaces, and for each subspace uploads *only one* randomly sample (i.e., dash shapes) to the central server. Then, the central server aggregates the sampled results from all devices and clusters them with a global SC, i.e., coloring the dash shapes. Lastly, each client device updates its clusters (i.e., coloring the local shapes) upon receiving the global clustering result.

device could distribute in a different number of subspaces and $L^{(z)}$ is unknown for each client device z. Applying the same number of clusters across these devices does not make much sense. Therefore, we propose to estimate the number of clusters on a device z according to local data distribution.

The number of clusters $r^{(z)}$ of the affinity graph $\mathbf{W}^{(z)}$ is estimated by finding the biggest spectral gap of normalized graph Laplacian $\mathbf{L}^{(z)}$, commonly referred to as eigengap heuristic [38]. Here, the normalized graph Laplacian $\mathbf{L}^{(z)}$ of graph $\mathbf{W}^{(z)}$ is defined by

$$\mathbf{L}^{(z)} = \mathbf{I}^{(z)} - \mathbf{D}^{(z)-\frac{1}{2}}\mathbf{W}^{(z)}\mathbf{D}^{(z)-\frac{1}{2}}$$

where $\mathbf{I}^{(z)} \in \mathbb{R}^{N^{(z)} \times N^{(z)}}$ is the identity matrix and $\mathbf{D}^{(z)} \in \mathbb{R}^{N^{(z)} \times N^{(z)}}$ is a diagonal matrix where $D_{ii} = \sum_{i=1}^{N} W_{i,j}$. The number of connected clusters $r^{(z)}$ is estimated by

$$r^{(z)} = \arg\max_{i \in [N^{(z)}-1]}(\sigma_{i+1} - \sigma_i), \tag{3}$$

where $\sigma_1 \le \sigma_2 \le \cdots \le \sigma_{N^{(z)}}$ are sorted singular values of normalized graph Laplacian $\mathbf{L}^{(z)}$.

Remark 1. As $r^{(z)}$ is the number of connected components of affinity graph $\mathbf{W}^{(z)}$, in practice, there may exist false connections due to errors in the sparse representations $\mathbf{C}^{(z)}$. Then, analyzing the eigenspectrum of the Laplacian $\mathbf{L}^{(z)}$ can be adopted to estimate the number of clusters due to its robustness against weak connections between the data points in different subspaces. We use the eigengap heuristic to estimate $r^{(z)}$ in the experiments of synthetic datasets. For the more complex high-dimensional datasets, we adopt a general upper bound of $r^{(z)}$ as described in Section VI, and achieve comparable performance in the experiments of real-world datasets.

Then, the normalized spectral clustering [24] is applied to the affinity graph $\mathbf{W}^{(z)}$ to generate the $r^{(z)}$ clusters $T^{(z)} = \{T_1^{(z)}, \dots, T_{r^{(z)}}^{(z)}\}$, $T_t^{(z)} \subseteq N^{(z)}$.

The underlying subspaces of $\mathbf{X}^{(z)}$ include the subspace spanned by the data points in each data partition $T_t^{(z)}$. We then estimate the bases of each subspace $\text{span}(\{\mathbf{x}_i^{(z)}\}_{i \in T_t^{(z)}})$. Specifically, data points $\{\mathbf{x}_i\}_{i \in T_t^{(z)}}$ are first arranged into a matrix $\mathbf{X}_{T_t^{(z)}}$ with a rank d_t, then the bases of $\text{span}(\{\mathbf{x}_i^{(z)}\}_{i \in T_t^{(z)}})$ can be recovered using the singular value decomposition (SVD)[3]. A basis denoted by $\mathbf{U}_{d_t}^{(z)}$ is obtained from the first d_t left singular vectors of $\mathbf{X}_{T_t^{(z)}}$.

C. Sampling Local Clusters

In order to upload the local clustering results, a natural approach is to transmit the set of basis $[\mathbf{U}_{d_t}^{(z)}]_{t=1}^{r^{(z)}}$, as is performed in k-means method k-FED where the local centroids are uploaded to the central server. However, this direct transmission of $[\mathbf{U}_{d_t}^{(z)}]_{t=1}^{r^{(z)}}$ exposes the local data structure and arises privacy concerns. Additionally, the high dimensionality of $\mathbf{X}^{(\mathbf{z})}$ results in high communication costs for the transmission of $[\mathbf{U}_{d_t}^{(z)}]_{t=1}^{r^{(z)}}$.

To mitigate information leakage and communication cost, while maintaining clustering effectiveness, we propose to randomly generate only one sample $\boldsymbol{\theta}_t^{(z)} \in \mathbb{R}^n$ to represent the local clustering result from each estimated subspace $\text{span}(\{\mathbf{x}_i^{(z)}\}_{i \in T_t^{(z)}})$.

$$\boldsymbol{\theta}_t^{(z)} \in \text{span}(\{\mathbf{x}_i^{(z)}\}_{i \in T_t^{(z)}}) \tag{4}$$

It is critical to design a sampling process to keep the utility of client devices, i.e., to ensure the central server can correctly aggregate and cluster the uploaded samples $[\boldsymbol{\Theta}^{(z)}]_{z=1}^{Z}$ from all the clients. Considering the requirement of the uniform data distribution to theoretically guarantee a successful SC, we propose to randomly generate a sample $\boldsymbol{\theta}_t^{(z)} \in \mathbb{R}^n$ uniformly distributed on the unit d_t-sphere in the d_t-dimensional subspace spanned by points in $T_t^{(z)}$. By drawing a coefficient vector $\boldsymbol{\alpha}_t^{(z)} \in \mathbb{R}^{d_t}$ from an isotropic Gaussian distribution $\boldsymbol{\alpha}_t^{(z)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the sampled vector $\boldsymbol{\theta}_t^{(z)}$ is generated by

$$\boldsymbol{\theta}_t^{(z)} = \frac{\mathbf{U}_{d_t}^{(z)}\boldsymbol{\alpha}_t^{(z)}}{\|\mathbf{U}_{d_t}^{(z)}\boldsymbol{\alpha}_t^{(z)}\|_2}. \tag{5}$$

Finally, each device z sends $r^{(z)}$ generated samples $\boldsymbol{\Theta}^{(z)} = [\boldsymbol{\theta}_t^{(z)}]_{t=1}^{r^{(z)}}$ to the central server. The entire process of local clustering and sampling (i.e., the key steps of Fed-SC) on every client device is outlined in Algorithm 2 and illustrated in Fig 2.

Remark 2. Fed-SC only requires one round of communication, with each local device z uploading randomly generated samples $\boldsymbol{\Theta}^{(z)}$, each uniformly distributed on a local subspace, to the central server. Precisely, $\boldsymbol{\Theta}^{(z)}$ consists of $r^{(z)}$ samples from $r^{(z)}$ local clusters, where each subspace is represented by a single randomly generated sample. Therefore, only a limited amount of local information is shared across the federated network. Besides, some existing privacy-preserving mechanisms such as holomorphic encryption [39] and differential

[3] Throughout the proposed algorithm, we use truncate SVD instead of standard SVD to reduce the computational complexity.

Algorithm 2 Local clustering and sampling

Input: Local data matrix $\mathbf{X}^{(z)}$ on client z

Solve the SSC optimization problem for each data point in $\mathbf{X}^{(z)}$;
Form an affinity graph $\mathbf{W}^{(z)}$ by $\mathbf{W}^{(z)} = |\mathbf{C}^{(z)}| + |\mathbf{C}^{(z)}|^T$;
Estimate the number of clusters $r^{(z)}$ in graph $\mathbf{W}^{(z)}$ by Eq. (3);
Apply spectral clustering to $\mathbf{W}^{(z)}$ to segment N points into $r^{(z)}$ clusters $T^{(z)} = (T_i^{(z)})_{t=1}^{r^{(z)}}$;
for $t \leftarrow 1$ **to** **do**
    Estimate the orthogonal basis $\mathbf{U}_{d_t}^{(z)}$ from data points in $T_t^{(z)}$;
    Sample the coefficient $\boldsymbol{\alpha}_t^{(z)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
    Generate the sample $\boldsymbol{\theta}_t^{(z)}$ by Eq. (5);
end for

Output: Data partitions $T^{(z)}$ and generated samples $\mathbf{\Theta}^{(z)} = [\boldsymbol{\theta}_1^{(z)}, \boldsymbol{\theta}_2^{(z)}, \ldots, \boldsymbol{\theta}_{r^{(z)}}^{(z)}]$

privacy [40] can be incorporated into Fed-SC to further protect the privacy while uploading $\mathbf{\Theta}^{(z)}$.

D. Central Clustering and Local Update

In this step, the central server receives the generated samples $[\mathbf{\Theta}^{(z)}]_{z=1}^{Z}$ from all devices. Note that each generated sample is distributed uniformly on the unit spheres in the subspaces estimated by all client devices, the received samples $[\mathbf{\Theta}^{(z)}]_{z=1}^{Z}$ are still consistent with the SC problem setting. Besides, the uniform distribution of $[\mathbf{\Theta}^{(z)}]_{z=1}^{Z}$ satisfies the uniform requirement of TSC on data distribution, thus also allowing the centralized server to run TSC on $[\mathbf{\Theta}^{(z)}]_{z=1}^{Z}$. Even though TSC can guarantee exact clustering, its ideal performance critically relies on the optimal choice of q and a sufficient number of points (see Fig. 4). Considering those limitations of TSC, the central server can also alternatively run SSC on $[\mathbf{\Theta}^{(z)}]_{z=1}^{Z}$ to obtain L clusters. For the global clustering result, we use $\tau_t^{(z)} \in [L]$ to denote the cluster assignments of the generated samples $\boldsymbol{\theta}_t^{(z)}$. The central server delivers the cluster assignments $\{\tau_t^{(z)}\}_{t=1}^{r^{(z)}}$ to each local device z.

Finally, each local device z receives the cluster assignments $\{\tau_t^{(z)}\}_{t=1}^{r^{(z)}}$ of $\mathbf{\Theta}^{(z)}$ and updates the partition $T^{(z)}$ into $\hat{T}^{(z)}$ according to $\{\tau_t^{(z)}\}_{t=1}^{r^{(z)}}$. By aggregating all the $\hat{T}^{(z)}$ from all devices, we have the final clustering $\hat{T}$ of the data that resides on all the devices in a federated network.

In the remaining, we use *Fed-SC (SSC)* and *Fed-SC (TSC)* to denote the Fed-SC methods where SSC and TSC are implemented at the central server, respectively.

E. Advantages of Fed-SC

We highlight the practical advantages of the proposed Fed-SC scheme as follows.

Communication efficiency. Like other one-shot federated learning methods [1], [41], Fed-SC just needs one round of communication, i.e., each device z sends the generated samples $\mathbf{\Theta}^{(z)} \in \mathbb{R}^{n \times r^{(z)}}$ to the server in the uplink channel and the server delivers the cluster assignments $\{\tau_t^{(z)}\}_{t=1}^{r^{(z)}}$ of $\mathbf{\Theta}^{(z)}$ to local device z in the downlink channel. Specifically, assuming each floating point value is quantized into q bits, then the bits of sending $[\mathbf{\Theta}^{(z)}]_{z=1}^{Z}$ in the uplink is $nq \sum_{z\in[Z]} r^{(z)}$. Since $\tau_t^{(z)} \in [L]$, the bits of downlink transmission is $\sum_{z\in[Z]} r^{(z)} \log L$. Since Fed-SC is one-shot, the total communication cost is final as above.

Scalability of large-size clustering. Compared to centralized SC algorithms, the proposed Fed-SC scheme works as a divide-and-conquer framework in which the large-scale dataset is split into local datasets of moderate size. Each local dataset can be efficiently clustered on each local device, and the clusters from different devices are aggregated at the central server. Particularly, we assume that a federated network contains Z devices and there are N points on each local device, making ZN points in total. The centralized spectral-based SC algorithms [27], [42] carry computational complexity of $O(Z^2N^2)$. As for Fed-SC, the SSC and sampling on each local device require the complexity of $O(N^2)$, while clustering on the server has a complexity that depends on the number of devices Z, i.e., $O(Z^2)$. The sequential and parallel running times of Fed-SC achieve reductions from $O(Z^2N^2)$ to $O(ZN^2 + Z^2)$ and $O(N^2 + Z^2)$, respectively.

Connectivity of affinity graph. Many works on SSC face a graph connectivity issue [43], i.e., the data points in the same subspace may form multiple connected components in an affinity graph that leads to an over-segmentation/over-clustering problem. However, Fed-SC by design can intrinsically mitigate this issue as each $\boldsymbol{\theta} \in \mathbf{\Theta}$ represents a group of data points in a connected component. The affinity graph on $\mathbf{\Theta}$ can induce a global affinity graph on $\mathbf{X}$ with more edges that mitigates over-segmentation. This advantage is validated empirically in the following experiments.

Robustness against communication noise. Owing to the robustness of SSC and TSC implemented at the central server together with our designed sampling procedures, Fed-SC can exhibit great robustness against communication noise. This advantage has been investigated in the experimental results.

V. Theoretical Guarantees

In this section, the effectiveness of the proposed federated SC scheme is verified theoretically. Our analysis assumes that the data are noiseless and normalized into the unit ℓ_2 norms. In particular, we focus on the following two common statistical data models: the *deterministic model* and the *semi-random model*.

Deterministic model. It is a generic statistical model without any stochastic assumptions on both the underlying subspaces and the distribution of data points. The raw data $[\mathbf{X}^{(z)}]_{z=1}^{Z}$ is inherently consistent with the deterministic model.

Semi-random model. The semi-random model assumes the data representations to be drawn i.i.d. uniformly in each subspace. In the proposed Fed-SC, the generated samples $[\mathbf{\Theta}^{(z)}]_{z=1}^{Z}$ are consistent with the semi-random model.

A. Definitions

Before we state the main theorems, we provide the following definitions of the quantities that characterize the arrangement of subspaces, the distribution of data points, and the affinity between subspaces.

Definition 1 (Subspace incoherence)**.** *Given a vector* $\mathbf{x}$ *and a matrix* $\mathbf{X}$*, let the* $\boldsymbol{\nu}(\mathbf{x}, \mathbf{X})$ *be the optimal solution to*

$$\max_{\boldsymbol{\nu} \in \mathbb{R}^n} \langle \mathbf{x}, \boldsymbol{\nu} \rangle, \quad s.t. \quad \|\mathbf{X}^T \boldsymbol{\nu}\|_\infty \leq 1.$$

We denote the dual direction $\boldsymbol{\nu}_{\ell,i}$ *of each point* $\mathbf{x}_i \in \mathbf{X}_\ell$ *as* $\boldsymbol{\nu}_{\ell,i} = \mathbb{P}_\ell \boldsymbol{\nu}(\mathbf{x}_{\ell,i}, \mathbf{X}_{\ell,-i}) / \|\mathbb{P}_\ell \boldsymbol{\nu}(\mathbf{x}_{\ell,i}, \mathbf{X}_{\ell,-i})\|_2$ *and arrange them as columns of* $\mathbf{V}_\ell$*. We define the subspace incoherence of a set of points* $\mathbf{X}_\ell$ *as* $\mu(\mathbf{X}_\ell) = \max_{\mathbf{x} \in \mathbf{X} \setminus \mathbf{X}_\ell} \|\mathbf{V}_\ell^T \mathbf{x}\|_\infty$.

Subspace incoherence characterizes the separability between a set of points in $\mathcal{S}_\ell$ and points in other subspaces. We illustrate this concept with the following examples.

Example 1 (Orthogonal subspace)**.** Suppose that the subspaces are orthogonal. Then, $\mu(\mathbf{X}_\ell) = 0$ for each $\ell \in [L]$ because each column of $\mathbf{V}_\ell$ is orthogonal to all other subspaces.

Example 2 (Disjoint subspace)**.** Suppose the subspaces are disjoint, i.e., $\dim(\mathcal{S}_\ell + \mathcal{S}_k) = \dim(\mathcal{S}_\ell) + \dim(\mathcal{S}_k)$ for each pair of subspaces. If the first canonical angle of $\mathcal{S}_\ell$ to one of following subspaces is $\pi/4$, then the subspace incoherence $\mu(\mathbf{X}_\ell) \geq \cos(\pi/4) = \frac{\sqrt{2}}{2}$.

Recalling the heterogeneous setting that the federated network is with statistical heterogeneity: for some $z \in [Z]$, $L^{(z)} < L$. With this notion of heterogeneity, on some device z, not all subspaces in $\{\mathcal{S}_\ell\}_{\ell=1}^L$ are presented. Hence, we need to characterize such data distribution among devices in federated networks by defining the *active set* for each subspace.

Definition 2 (Active set)**.** *For each subspace* $\mathcal{S}_\ell, \ell \in [L]$*, the active set of subspace* $\alpha(\ell)$ *is defined as,*

$$\alpha(\ell) = \{k \in [L] : \textit{if there exists at least one device containing data points distributed on both } S_\ell \textit{ and } S_k,\ k \neq \ell.\}$$

If the data points from some subspaces are never involved on any device simultaneously, there is a relaxation of the subspace incoherence requirement which only depends on the subspaces existing in its active set. So, we define *active subspace incoherence*, which will be used to characterize the relaxation of subspace incoherence requirement in federated networks with statistical heterogeneity.

Definition 3 (Active subspace incoherence)**.** *For a set of points* $\mathbf{X}_\ell$ *with its active sets of subspaces* $\alpha(\ell)$*, we define the active subspace incoherence as*

$$\tilde{\mu}(\mathbf{X}_\ell) = \max_{\mathbf{x} \in \mathbf{X}_{\alpha(\ell)}} \|\mathbf{V}_\ell^T \mathbf{x}\|_\infty$$

where $\mathbf{X}_{\alpha(\ell)} = [\mathbf{X}_k]_{k \in \alpha(\ell)}$ *denotes the subspaces indexed in the active set of* $\mathcal{S}_\ell$.

Definition 4 (Inradius)**.** *The inradius of a convex body* $\mathcal{P}$*, denoted by* $r(\mathcal{P})$*, is defined as the radius of the largest Euclidean ball inscribed in* $\mathcal{P}$.

The inradius of $\mathcal{P}(X_{\ell,-i})$ measures the distribution of data points on a subspace. The small inradius implies that the data points have a skewed distribution on the subspace, as illustrated in Fig. 3.

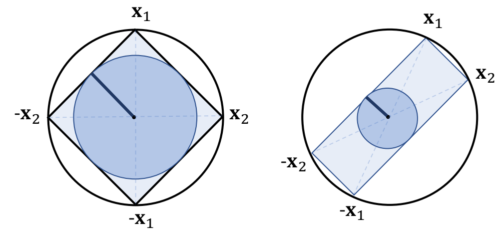

Fig. 3. Illustration of inradius: well-dispersed distribution of data has a large inradius, and the skewed distribution of data has a small inradius.

For the semi-random model, due to the points being distributed uniformly in each subspace, the theoretical analysis only depends on the distance between subspaces. The subspace distance is measured by the affinity between two subspaces as follows.

Definition 5 (Affinity between subspaces)**.** *The affinity between two subspaces is defined by:*

$$\operatorname{aff}(\mathcal{S}_k, \mathcal{S}_\ell) = \sqrt{\cos^2 \phi_{k\ell}^{(1)} + \cos^2 \phi_{k\ell}^{(2)} + \cdots + \cos^2 \phi_{k\ell}^{(d_k \wedge d_\ell)}}$$

where $\phi_{k\ell}^{(i)}$ *is the* i*-th canonical angle between subspaces* $\mathcal{S}_k$ *and* $\mathcal{S}_\ell$.

For the further analysis of theoretical guarantee, we consider a condition for the distribution of data points on a subspace, called *general position*. The assumption that data points are in general position is mild and widely used in analyses of GPCA [44] and SC post-processing [45].

Definition 6 (General position)**.** *Given a data matrix* $\mathbf{X} = [\mathbf{X}_\ell]_\ell^L$*, we say* $\mathbf{X}$ *is in general position if for all* $k \leq d_\ell$*, any subset of* k *points (columns) in* $\mathbf{X}_\ell$ *are linearly independent.*

B. Main Theorems

Although centralized SC methods, such as SSC and TSC, have been proven effective in seminal works [10], [17], the proposed one-shot federated scheme, Fed-SC, requires a new theoretical foundation to guarantee clustering effectiveness. Fed-SC comprises steps of local and central clustering, which differ significantly in their data and statistical models, as summarized in Table II. Fed-SC's efficacy relies on successful executions of both local and central clustering. To establish the effectiveness in a federated network, we present theorems with sufficient conditions for both local clustering (SSC) and central clustering (SSC/TSC) under which Fed-SC can meet the criteria of effective clustering (i.e., SEP or exact clustering).

Theorem 1 (Fed-SC (SSC))**.** *Assume that each* $\mathbf{X}^{(z)}$ *is in general position and the non-zero* $N_\ell^{(z)} \geq d_\ell + 1$ *for all* $\ell \in [L]$ *and* $z \in [Z]$*. Let* $r' = \max_{z \in [Z]} r^{(z)}$, $N'_\ell = \min\{N_\ell^{(z)} | N_\ell^{(z)} >$

TABLE II
SUMMARY OF LOCAL AND CENTRAL CLUSTERING IN FED-SC.

	Local clustering	Central clustering
Data	$\mathbf{X}^{(z)}$	$[\mathbf{\Theta}^{(z)}]_{z=1}^{Z}$
Statistical model	deterministic model	semi-random model
Algorithm	SSC	SSC/TSC
Criteria	SEP	SEP/exact clustering

$0\}_{z\in[Z]}$ and $\mathbb{W}_\ell^{N'_\ell}$ be the set of all submatrices of X_ℓ with N'_ℓ columns. If for each $\ell \in [L]$, the active deterministic condition

$$\min_{\tilde{\mathbf{X}}_\ell \in \mathbb{W}_\ell^{N'_\ell}} \min_{i:x_i\in\tilde{\mathbf{X}}_\ell} r(\mathcal{P}(\tilde{\mathbf{X}}_{\ell,-i})) > \tilde{\mu}(\mathbf{X}_\ell),$$

and the global semi-random condition

$$c\sqrt{\log\frac{Z_\ell - 1}{d_\ell}} > \max_{k:k\neq l}\ t\log[Lr'Z_\ell(r'Z_k+1)]\frac{\mathrm{aff}(\mathcal{S}_\ell,\mathcal{S}_k)}{\sqrt{d_k}}$$

are satisfied, then the final clustering of Fed-SC (SSC) holds SEP with probability at least $1-\sum_{\ell=1}^{L} Z_\ell e^{-\sqrt{d_\ell}\sqrt{Z_\ell-1}} - \frac{1}{L^2}\sum_{k\neq\ell}\frac{e^{-t/4}}{Z_\ell(Z_k+1)}$.

Theorem 1 shows that if different subspaces in the active sets are separated and the points of each subspace are well-dispersed (i.e., satisfying active deterministic condition), and the subspaces are not close to each other (i.e., satisfying global semi-random condition), then Fed-SC (SSC) can correctly detect all subspaces (i.e., holding SEP) and thus successfully cluster the data with high probability. It should be noted that both these conditions include the special cases of orthogonal subspace (Example 1) and disjoint subspace (Example 2), which have been widely investigated in [9], [10], [17]. Theorem 1 explains quite precisely when the proposed Fed-SC (SSC) is successful and provides insights on what kind of subspace property and data distribution can lead to good clustering performance. Fed-SC (SSC) can be applied to general datasets, and the expected performance on synthetic and real-world datasets further verifies its effectiveness.

Note that the subspaces not contained in the active set of each other (i.e., $\mathcal{S}_\ell$ and $\mathcal{S}_k, k\notin\alpha(\ell)$) are only required to satisfy the global semi-random condition. If the federated network is with more statistical heterogeneity, the cardinality of active sets will be much smaller, and then less subspaces are required to satisfy the active deterministic condition. Furthermore, the semi-random condition can be relaxed by the statistical heterogeneity. To illustrate that, we assume the subspaces are of the same dimension and there are the same number of devices containing data points from each subspace, i.e., $d \triangleq d_\ell$ and $Z' \triangleq Z_\ell, \forall\ell\in[L]$. We formally state the relation between Z' and the upper bound of the affinity between subspaces in the following corollary.

Corollary 1. *Suppose that $d_\ell = d$ and $Z_\ell = Z', \forall\ell\in[L]$. A set of subspaces $\{\mathcal{S}_\ell\}_{\ell=1}^{L}$ satisfy the global semi-random condition for Fed-SC (SSC) if*

$$\max_{k,\ell:k\neq\ell}\mathrm{aff}(\mathcal{S}_\ell,\mathcal{S}_k) < \frac{c\sqrt{d\log\frac{Z'-1}{d}}}{t\log[Lr'Z'(r'Z'+1)]}.$$

Ignoring the constants, the upper bound of affinity between subspaces is $\Omega(\frac{\sqrt{d}}{\sqrt{\log Z'}} - \frac{\sqrt{d\log d}}{\log Z'})$ for Fed-SC (SSC). It is easy to see that the upper bound becomes higher as Z' decreases when d is small. So for each $\ell\in[L]$, the affinity requirement for $\mathcal{S}_\ell$ and $\mathcal{S}_k, k\notin\alpha(\ell)$, becomes weaker as Z' get smaller, verifying the benefit of our notion of statistical heterogeneity[4].

The theorem for Fed-SC (TSC) is introduced as follows.

Theorem 2 (Fed-SC (TSC))**.** *Assume that each $\mathbf{X}^{(z)}$ is in general position and the non-zero $N_\ell^{(z)} \geq d_\ell+1$ for all $\ell\in[L]$ and $z\in[Z]$. Let $r' = \max_{z\in[Z]} r^{(z)}$, $N'_\ell = \min\{N_\ell^{(z)} | N_\ell^{(z)} > 0\}_{z\in[Z]}$ and $\mathbb{W}_\ell^{N'_\ell}$ be the set of all submatrices of X_ℓ with N'_ℓ columns. Suppose that TSC chooses the parameter $q\in[c_1\log(r'\max_\ell Z_\ell), \min_\ell Z_\ell/6]$ with $c_1 = 18(12\pi)^{\max_\ell d_\ell - 1}$. If for each $\ell\in[L]$, the active deterministic condition*

$$\min_{\tilde{\mathbf{X}}_\ell \in \mathbb{W}_\ell^{N'_\ell}} \min_{i:x_i\in\tilde{\mathbf{X}}_\ell} r(\mathcal{P}(\tilde{\mathbf{X}}_{\ell,-i})) > \tilde{\mu}(\mathbf{X}_\ell)$$

and the global semi-random condition

$$\max_{\ell,k:k\neq\ell}\frac{\mathrm{aff}(\mathcal{S}_\ell,\mathcal{S}_k)}{\sqrt{d_\ell\wedge d_k}} \leq (15\log\sum_{\ell\in[L]} r'Z_\ell)^{-1}$$

are satisfied, then Fed-SC (TSC) clusters $\mathbf{X}$ exactly with the probability at least $1-\frac{10}{\sum_{\ell\in[L]}Z_\ell} - \sum_{\ell\in[L]}(Z_\ell e^{-c(Z_\ell-1)} + 2Z_\ell^{-2})$.

The interpretation of Theorem 2 is analogous to that of Theorem 1 except that Fed-SC (TSC) is guaranteed with exact clustering. It should be noted that Z_ℓ is required to be exponential with respect to d_ℓ because of the exponential dependency of c_1 as in [10]. This restriction on Z_ℓ and d_ℓ limits the scope of applications especially when d_ℓ is large or there are no sufficient devices in federated networks (see experiments in VI-A). Similar to Fed-SC (TSC), there also exists a relation between Z_ℓ and the upper bound of affinity between subspaces. With the same settings in Corollary 1, we state this formally in Corollary 2.

Corollary 2. *Suppose that $d_\ell = d$ and $Z_\ell = Z', \forall\ell\in[L]$. A set of subspaces $\{\mathcal{S}_\ell\}_{\ell=1}^{L}$ satisfy the global semi-random condition for Fed-SC (TSC) if*

$$\max_{k:k\neq\ell}\mathrm{aff}(\mathcal{S}_\ell,\mathcal{S}_k) \leq \frac{\sqrt{d}}{15\log(Lr'Z')}.$$

Analogue to Corollary 1, the upper bound of the affinity between subspaces is $\Omega(\frac{\sqrt{d}}{\log Z'})$ for Fed-SC (TSC) and statistical heterogeneity can relax the upper bounds between subspaces $\mathcal{S}_\ell$ and $\mathcal{S}_k, k\notin\alpha(\ell)$.

[4]As described in Section III-B, Z_ℓ and $L^{(z)}$ characterize the data partition across the federated network, and there is an equivalence between Z_ℓ and $L^{(z)}$ established by $\sum_{z\in[Z]} L^{(z)} = \sum_{\ell\in[L]} Z_\ell$. Thus, the federated network is also with statistical heterogeneity if $Z_\ell < Z$ for some $\ell\in[Z]$.

C. Proof of Main Theorems

In this section, we use Lemmas 1, 2 and 3 to establish the proof of Theorem 1, and use Lemmas 1, 2 and 4 to establish the proof of Theorem 2. Specifically, we lay out the steps of the proof for main theorems as follows.

1) We firstly show that the active deterministic condition is sufficient for local data to hold the SEP and state this in Lemma 1.
2) Then, we make use of Lemma 2 stating that the subspace spanned by the points from a connected component is identical to one of the subspaces $\{\mathcal{S}_\ell\}_\ell^L$.
3) Finally, the global semi-random conditions are sufficient for generated samples to hold SEP or exact clustering and we state this in Lemmas 3 and 4, respectively.

Lemma 1. *Let $\mathbf{X} = [\mathbf{X}_\ell]_{\ell=1}^L$ be the data points drawn from $\bigcup_{\ell=1}^L \mathcal{S}_\ell$ and any non-empty subset of local data $\mathbf{X}_\ell^{(z)} \subseteq \mathbf{X}^{(z)}$ contains at least N'_ℓ data points in each local device $z \in [Z]$. Let $\mathbb{W}_\ell^N$ be the set of the submatrix of X_ℓ with N data points. If for each $l \in [L]$,*

$$\min_{\tilde{\mathbf{X}}_\ell \in \mathbb{W}_\ell^{N'_\ell}} \min_{i:x_i \in \tilde{\mathbf{X}}_\ell} r(\mathcal{P}(\tilde{\mathbf{X}}_{\ell,-i})) > \tilde{\mu}(\mathbf{X}_\ell),$$

then the local SSC holds SEP for each $\mathbf{X}^{(z)}$ in local device $z \in [Z]$.

Proof. From Theorem 2.5 in [17], the sufficient condition on local data $\mathbf{X}_\ell^{(z)}$ for SSC to hold SEP is

$$\min_{i:x_i \in \mathbf{X}_\ell^{(z)}} r(\mathcal{P}(\mathbf{X}_{\ell,-i}^{(z)})) > \mu(\mathbf{X}_\ell^{(z)}). \tag{6}$$

Since $N'_\ell = \min\{N_\ell^{(z)} | N_\ell^{(z)} > 0\}_{z \in [Z]}$, there exists a matrix $\tilde{\mathbf{X}}_\ell \in \mathbb{W}_\ell^{N'_\ell}$ of which the columns are a subset of the columns of $\mathbf{X}_\ell^{(z)}$, $\tilde{\mathbf{X}}_\ell \subseteq \mathbf{X}_\ell^{(z)}$, for any local data $\mathbf{X}_\ell^{(z)}, \forall z \in [Z]$. $\min\limits_{i:x_i \in \mathbf{X}_\ell^{(z)}} r(\mathcal{P}(\mathbf{X}_{\ell,-i}^{(z)}))$ for each $z \in [Z]$. In summery, for each $\ell \in [L]$ and $z \in [Z]$

$$\min_{\tilde{\mathbf{X}}_\ell \in \mathbb{W}_\ell^{N'_\ell}} \min_{i:x_i \in \tilde{\mathbf{X}}_\ell} r(\mathcal{P}(\tilde{\mathbf{X}}_{\ell,-i})) \leq \min_{i:x_i \in \mathbf{X}_\ell^{(z)}} r(\mathcal{P}(\mathbf{X}_{\ell,-i}^{(z)})). \tag{7}$$

On the other hand, the matrix of local projected dual directions $V_\ell^{(z)}$ is a subset of V_ℓ, so $\|{V_\ell^{(z)}}^T \mathbf{x}\|_\infty \leq \|V_\ell^T \mathbf{x}\|_\infty, \forall \mathbf{x} \in \mathbb{R}^n$. For each $\ell \in [L]$, data points from any subspace indexed by k, $k \notin \alpha(\ell)$ is not contained on any local devices. Conversely, any data point on each device that contains data points from subspace $\mathcal{S}_\ell$ is distributed on either subspace $\mathcal{S}_\ell$ or the subspaces in its active set $\alpha(\ell)$, that is, $\mathbf{X}^{(z)} \setminus \mathbf{X}_\ell^{(z)} \subseteq \mathbf{X}_{\alpha(\ell)}$. We then have inequality,

$$\begin{aligned}
\tilde{\mu}(\mathbf{X}_\ell) = \max_{\mathbf{x} \in \mathbf{X}_{\alpha(l)}} \|V_\ell^T \mathbf{x}\|_\infty &\geq \max_{\mathbf{x} \in \mathbf{X}^{(z)} \setminus \mathbf{X}_\ell^{(z)}} \|V_\ell^T \mathbf{x}\|_\infty \\
&\geq \max_{\mathbf{x} \in \mathbf{X}^{(z)} \setminus \mathbf{X}_\ell^{(z)}} \|{V_\ell^{(z)}}^T \mathbf{x}\|_\infty \qquad (8) \\
&= \mu(\mathbf{X}_\ell^{(z)}). \qquad (9)
\end{aligned}$$

The combination of equations (7) and (9) concludes the proof. □

Lemma 2. *If data points $\mathbf{X}^{(z)}$ are in general position, non-zero $N_\ell^{(z)} \geq d_\ell + 1$ and local SSC holds SEP, the subspace spanned by the points in each connected component of the local affinity graph is identical to one of the underlying subspaces $\{\mathcal{S}_\ell\}_\ell^L$.*

Proof. Given a affinity graph $\mathbf{W}^{(z)}$ built by SSC , suppose there are K connected components $\mathbf{W}_k^{(z)}, k \in [K]$. By SEP, the data points are distributed on the same subspace if they are connected. Hence, there must exist an underlying subspace $\mathcal{S}_\ell$ for each connected component $\mathbf{W}_k$. If $\mathbf{X}$ is in general position and non-zero $N_\ell^{(z)} \geq d_\ell + 1$ then the number of points in $\mathbf{W}_k^{(z)}$ larger than or equal to $d_\ell + 1$, which can be concluded by that for each $\mathbf{x}_i \in \mathbf{X}_\ell$, the optimal solution for optimization problem in SSC is the linear combination of other linear independent d_ℓ points. Furthermore, the linearly independent data points in V_k span the exact subspace $\mathcal{S}_\ell$. □

With Lemmas 1 and 2, the estimated subspaces are the same as the exact underlying subspaces. Recalling the proposed sampling method in Fed-SC such that the samples $\mathbf{\Theta}$ generated from subspaces are uniformly distributed on the unit spheres. This sampling procedure is consistent with the semi-random model. Next, we introduce sufficient conditions for $\mathbf{\Theta}$ to hold SEP or exact clustering by bounding the number of samples from each subspace, according to the data partition over the federated network.

Lemma 3. *Let $\mathbf{\Theta} = [\mathbf{\Theta}_\ell]_{\ell=1}^L$ be the data points drawn from $\bigcup_{\ell=1}^L \mathcal{S}_\ell$ in semi-random model and $\mathbf{\Theta}_\ell \in \mathbb{R}^{n \times N_\ell}$ with $Z_\ell \leq N_\ell \leq r' Z_\ell$. If*

$$\max_{k:k \neq l} t(\log[r' Z_\ell (r' Z_k + 1)] + \log L) \frac{\mathrm{aff}(\mathcal{S}_\ell, \mathcal{S}_k)}{\sqrt{d_k}} < c \sqrt{\log \frac{Z_\ell - 1}{d_\ell}}$$

for each $\ell \in [L]$, then SSC holds SEP for $\mathbf{\Theta}$ with probability at least $1 - \sum_{\ell=1}^L Z_\ell e^{-\sqrt{d_\ell}\sqrt{Z_\ell - 1}} - \frac{1}{L^2} \sum_{k \neq \ell} \frac{e^{-t/4}}{Z_\ell(Z_k+1)}$.

Proof. Identical to the proof of Lemma 1, this lemma can be concluded by scaling the inequality of Theorem 2.8 in [17]. □

By the Lemma 3, we know that the generated samples $\theta_t^{(z)}$ from the subspace $\mathcal{S}_\ell$ are labeled correctly by $\tau_t^{(z)} = \ell$ with corresponding high probability. Recalling the local update as we described in Algorithm 1, local data points from different subspaces are partitioned into different clusters, then Fed-SC (SSC) holds SEP for the global data. Thus, We conclude the proof of Theorem 1.

Lemma 4. *Let $\mathbf{\Theta} = [\mathbf{\Theta}_\ell]_{\ell=1}^L$ be the data points drawn from $\bigcup_{\ell=1}^L \mathcal{S}_\ell$ in semi-random model and $\mathbf{\Theta}_\ell \in \mathbb{R}^{n \times N_\ell}$ with $Z_\ell \leq N_\ell \leq r Z_\ell$. Suppose TSC chooses parameter $q \in [c_1 \log(r \max_\ell Z_\ell), \min_\ell Z_\ell / 6]$ with $c_1 = 18(12\pi)^{\max_\ell d_\ell - 1}$. If*

$$\max_{l,k:k \neq l} \frac{\mathrm{aff}(\mathcal{S}_\ell, \mathcal{S}_k)}{\sqrt{d_\ell \wedge d_k}} \leq (15 \log \sum_{l \in [L]} r Z_\ell)^{-1},$$

then TSC can correctly cluster the data $\mathbf{X}$ *with the probability at least* $1 - \frac{10}{\sum_{\ell \in [L]}} Z_\ell - \sum_{\ell \in [L]} (Z_\ell e^{-c(Z_\ell - 1)} + 2Z_\ell^{-2})$.

Proof. Identical to the proof of Lemma 1, this lemma can be concluded by scaling the inequality of Theorem 2 in [10]. □

Analog to the proof of Theorem 1, local data points from the same subspace are clustered into the same cluster by combining the Lemmas 1, 2 and 4. Thus, the Fed-SC (TSC) can cluster the data on federated networks with corresponding high probability, which concludes the Theorem 2.

VI. Experiments

In this section, we mainly demonstrate the effectiveness of the proposed Fed-SC method by testing on both synthetic and real data. All the experiments are performed on a machine with an Intel Xeon(R) 2.6GHz CPU and 502 GB main memory.

Datasets. We demonstrate the clustering performance of the proposed Fed-SC on synthetic data and real-world datasets including Extended MNIST (EMNIST) [46] and Columbia Object Image Library (COIL100) [47]. The datasets and their preprocessing steps are outlined as follows:

- We randomly generate L subspaces (adjustable) each of the same dimension $d = 5$ by drawing i.i.d. orthonormal basis matrices in $\mathbb{R}^{20}$. The synthetic data is obtained by multiplying random gaussian coefficients with each basis matrix.
- EMNIST contains a set of handwritten character digits, with 814,255 characters of 62 unbalanced classes. We resize each image into a size of 32×32 and compute the features by using a scattering convolution network [48]. Then, we concatenate the features into a 3472-dimensional vector for each image.
- COIL100 contains 100 different objects each with 72 images taken at different pose intervals. Data augmentations including random brightness and contrast changes are applied to obtain an augmented dataset with size over $60,000$. We further convert the augmented images into gray-scale, resize them to the size of 32×32, and concatenate the pixels of each processed image into a vector with dimension 1024.

In the following experiments, we set up Z devices and randomly distribute the data among Z devices such that each device z receives data points from $L' \leq L$ clusters.

Baselines and implementation details. We use state-of-the-art centralized SC and FC methods as baselines. The centralized methods includes: SSC [9], NSN [27], TSC [10], SSCOMP [42] and EnSC [26]. The state-of-the-art one-shot federated clustering method is k-FED [1].

We implement the Lasso optimization algorithm of SSC with the sparse modeling software (SPAMS) [49] instead of the Alternating Direction Method of Multipliers (ADMM) [50]. The parameter λ of the optimization for $\mathbf{x}_i$ is determined by $\lambda = \max_{j \neq i} |\mathbf{x}_j^T \mathbf{x}_i|/50$, as presented in Proposition 1 of [9]. We set $q = \max(3, \lceil Z/L \rceil)$ for the TSC implementation in the Fed-SC (TSC) scheme and $\max(3, \lceil N/100L \rceil)$ for the centralized TSC algorithm. In the proposed federated scheme, Fed-SC, we take the upper bound of $r^{(z)}$ with $r^{(z)} = \max_{z \in [Z]} L^{(z)}$ and target dimension $d_t = 1$ for the experiments of real-world datasets. For the greedy baseline algorithms NSN and orthogonal matching pursuit (OMP) [42], we use the faster implementation for NSN [27] and an optimized OMP for SC [42], respectively. Moreover, we use the scalable oracle-based active set methods for EnSC [26]. The affinity matrices built by all the above algorithms are stored as sparse matrices, which can be efficiently computed.

Evaluation metrics and methodology. All algorithms are evaluated by the *clustering accuracy* (ACC: $a\%$) [42] and *normalized mutual information* (NMI: $n\%$) [51]. ACC is computed by finding the best label alignment of clustering result over all possible permutations as follows,

$$a = \max_{\pi} \frac{100}{N} \sum_{i \in [L], j \in [N]} \tilde{Q}_{\pi(i),j} Q_{i,j} \tag{10}$$

where $Q, \tilde{Q} \in \{0,1\}^{L \times N}$ are the ground-truth and estimated data labeling matrices, respectively, and $\pi(\cdot)$ is the permutation of L clusters. NMI evaluates the certainty of clustering results about the ground-truth class labels with a scale between 0 and 100 in percentage. It is calculated by:

$$n = 100 \frac{2\mathrm{MI}(\tilde{Q}; Q)}{\mathrm{H}(\tilde{Q}) + \mathrm{H}(Q)} \tag{11}$$

where $\mathrm{H}(\cdot)$ and $\mathrm{MI}(\cdot;\cdot)$ denote the entropy and mutual information between two label assignments, respectively.

In addition, the connectivity of the affinity graph is evaluated. Let $\lambda_l^{(i)}$ denote the i-th eigenvalue of the normalized affinity graph Laplacian corresponding to $\mathcal{S}_\ell$. The metrics of *connectivity* (CONN: c) is computed by $c = \min\{\lambda_l^{(2)}\}_{\ell \in L}$ and the average quantity $\bar{c} = \frac{1}{L} \sum_{l \in [L]} \lambda_l^{(2)}$ [52]. For the efficiency evaluation of federated schemes, the running time T is measured by $\mathrm{T} = \sum_{z \in [Z]} \mathrm{T}^{(z)} + \mathrm{T}_c$ where $\mathrm{T}^{(z)}$ is the running time on client device z and T_c is the time on the central server.

Evaluation is first conducted on synthetic data with more control of data distributions to exhibit multifaceted comparisons. Then, we empirically evaluate the effectiveness and efficiency of Fed-SC on real-world datasets. Both evaluations have practically witnessed the benefit of statistical heterogeneity that boosts the clustering performance.

A. Evaluation on Synthetic Data

Effectiveness of Fed-SC. We evaluate the clustering performance of Fed-SC and k-FED on data from two different partitions among Z devices: 1) random IID partitions by setting $L' = L = 20$, denoted as IID, and 2) non-IID partitions by setting $L' \in \{2, 10\}$, denoted as Non-IID-2 and Non-IID-10, respectively. By varying Z in a wide range of 200 to 2000, the clustering results of Fed-SC (SSC), Fed-SC (TSC), and k-FED are depicted in Fig. 4. We observe that both Fed-SC (SSC) and Fed-SC (TSC) outperform k-FED in accuracy and normalized mutual information with a significant increase. As

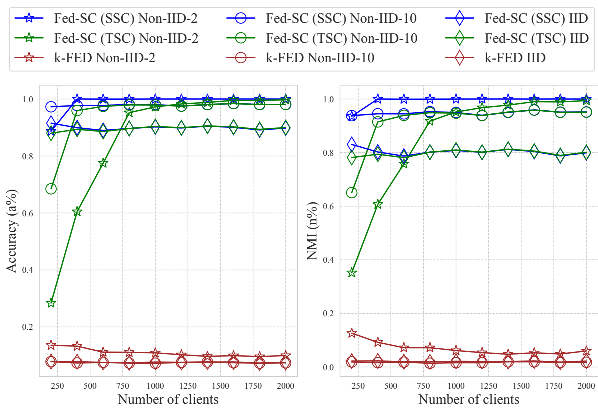

Fig. 4. The clustering accuracy and normalized mutual information of federated clustering methods, Fed-SC (SSC), Fed-SC (TSC), and k-FED, as functions of the number of devices with the different fashion of data partition.

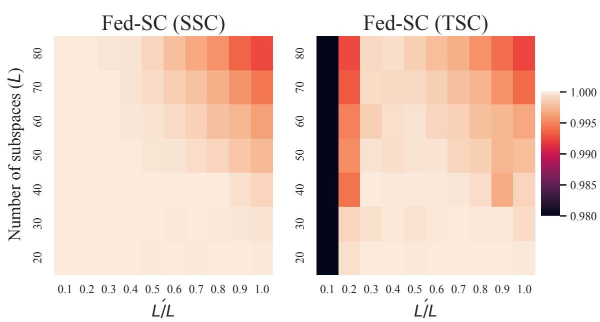

Fig. 5. The clustering accuracy of Fed-SC (SSC) and Fed-SC (TSC) as functions of L'/L and the number of subspaces L. Brighter cells represent higher accuracy of clustering.

we discussed in Theorem 2, the requirement of the large value of Z_ℓ degrades the clustering performance when the value Z is small but eventually has the same good performance as Fed-SC (SSC). Besides, we note that all the federated methods achieve better performance for non-IID data, which verifies the benefits of heterogeneity.

To further demonstrate the benefits of heterogeneity for Fed-SC, we set $Z = 400$ and conduct the experiments by varying the number of subspaces L and the ratio of L' to L. The corresponding results of Fed-SC (SSC) and Fed-SC (TSC) are depicted in Fig. 5. We observe that the accuracy of decreases as the values of L'/L and L get larger. For Fed-SC (TSC), the very small L' badly affects the accuracy because TSC requires a sufficient number of samples as we discussed in Theorem 2. Even still, the obvious decrease in accuracy of Fed-SC (TSC) still exists in the range from 0.3 to 1.0 for L'/L. The experimental results demonstrate that Fed-SC can achieve better performance if $L' \ll L$, verifying the benefits of heterogeneity.

We also focus on comparing Fed-SC's performance with the centralized SC methods in statistically heterogeneous federated networks. Specifically, we set $L = 50$ (a middle value from the previous experiment in Fig. 5) and $L' = 3$

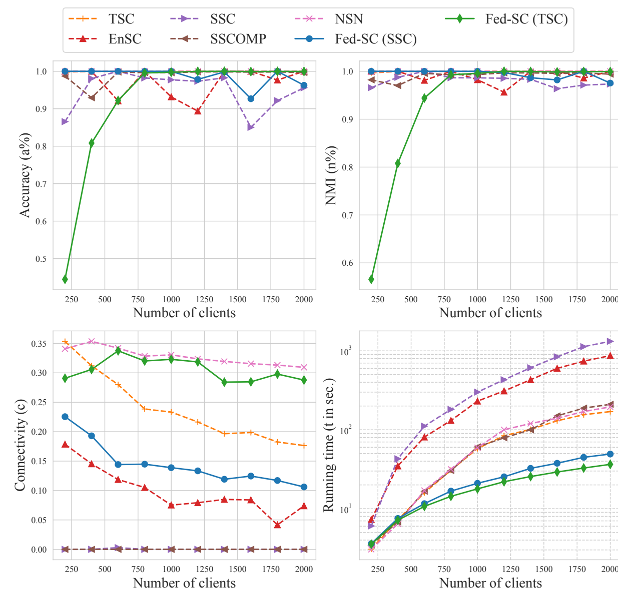

Fig. 6. The performance of Fed-SC(SSC), Fed-SC(TSC), and the centralized SC algorithms on synthetic data.

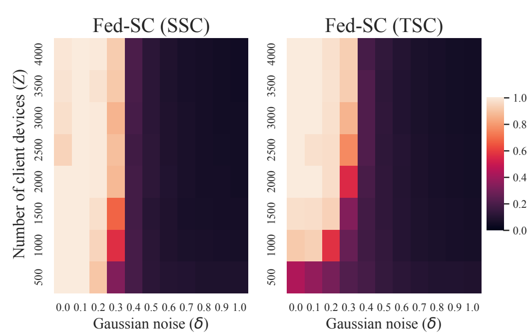

Fig. 7. The clustering accuracy of Fed-SC (SSC) and Fed-SC (TSC) as functions of δ and the number of devices Z. Brighter cells represent higher accuracy of clustering.

to ensure heterogeneity ($L' \ll L$). Fig. 6 illustrates each metric of evaluated methods as a function of Z. It can be seen that Fed-SC (SSC) achieves leading accuracy compared with centralized methods and Fed-SC (TSC) gets higher accuracy with a growth of Z. Meanwhile, the result shows that Fed-SC improves connectivity compared to centralized SSC and TSC methods. This is because the affinity graph on generated samples $\mathbf{\Theta}$ can induce a denser global affinity graph on $\mathbf{X}$.

Efficiency of Fed-SC. As depicted in the last plot of Fig. 6, Fed-SC can significantly reduce the time cost compared to the centralized SC methods, especially when the number of local devices Z is large. This corresponds to the scalability analysis in Section IV-E, the sequential runtime of Fed-SC reduces from $O(Z^2N^2)$ of centralized algorithms to $O(ZN^2 + Z^2)$, where N is the number of local data points.

Robustness to communication noise. We next study the robustness of the proposed Fed-SC to communication noise. On each device z, the generated samples $\Theta^{(z)}$ are subjected

TABLE III
PERFORMANCE COMPARISON ON EMNIST AND CIFAR-10 WHERE '-' DENOTES THE METRIC CANNOT BE COMPUTED PROPERLY. *: THE RUNNING TIME OF SSC FOR EMNIST EXCEEDS THE TIME LIMIT OF 1 DAY.

EMNIST ($2 \leq L^{(z)} \leq 4, z \in [Z]$)				
Methds	ACC($a\%$)	NMI($n\%$)	CONN($\bar{c}$)	T(sec.)
Fed-SC (SSC)	<u>85.77</u>	**88.28**	0.0019	262.83
Fed-SC (TSC)	**86.17**	<u>87.00</u>	0.0186	237.31
k-FED	56.68	67.18	-	<u>16.00</u>
k-FED + PCA-10	11.47	31.23	-	**7.95**
k-FED + PCA-100	11.64	31.28	-	16.18
SSC*	-	-	-	-
SSCOMP	56.17	70.26	0.000	12943.46
EnSC	60.83	74.00	<u>0.0317</u>	29459.42
TSC	49.04	66.92	0.0131	2511.73
NSN	41.68	63.82	**0.1571**	8117.37
Augmented COIL100 ($2 \leq L^{(z)} \leq 4, z \in [Z]$)				
Methds	ACC($a\%$)	NMI($n\%$)	CONN($\bar{c}$)	T(sec.)
Fed-SC (SSC)	**74.43**	**85.09**	0.0104	96.65
Fed-SC (TSC)	<u>57.54</u>	75.24	0.0579	78.12
k-FED	31.52	52.05	-	<u>3.03</u>
k-FED + PCA-10	8.59	26.18	-	**1.44**
k-FED + PCA-100	8.43	26.44	-	3.64
SSC	45.25	71.93	0.0006	31676.33
SSCOMP	41.17	68.26	0.0118	1616.64
EnSC	51.55	76.91	0.0324	3842.41
TSC	53.06	<u>78.99</u>	<u>0.1859</u>	809.27
NSN	30.46	46.97	**0.4280**	1765.18

TABLE IV
CLUSTERING ACCURACIES ($a\%$) WITH DIFFERENT NUMBER OF LOCAL CLUSTERS L'

EMNIST					
L'	2	4	6	8	10
Fed-SC (SSC)	**88.96**	**82.74**	**75.58**	**72.66**	**69.76**
Fed-SC (TSC)	<u>86.03</u>	<u>81.37</u>	<u>71.95</u>	<u>69.24</u>	<u>65.57</u>
k-FED	67.70	57.25	46.56	38.19	25.29
k-FED + PCA-10	13.41	9.02	7.62	7.82	7.14
k-FED + PCA-100	13.13	9.39	7.93	7.61	7.19
Augmented COIL100					
L'	2	4	6	8	10
Fed-SC (SSC)	**82.07**	**72.44**	**49.15**	**45.83**	**39.31**
Fed-SC (TSC)	<u>75.33</u>	<u>66.54</u>	<u>47.99</u>	<u>44.48</u>	<u>38.09</u>
k-FED	37.08	25.56	19.60	19.12	17.88
k-FED + PCA-10	10.78	7.01	5.40	5.56	5.61
k-FED + PCA-100	11.40	7.08	5.54	5.84	5.47

to the Gaussian noise with variance $\frac{\delta}{\sqrt{r^{(z)}}}$ to simulate the communication noise. We vary δ and the number of devices Z. The result, depicted in Fig. 7, shows that Fed-SC is robust to communication noise.

B. Empirical Evaluation on Real World Datasets

In this section, we empirically evaluate the clustering performance of Fed-SC for high-dimensional real-world datasets over heterogeneous federated networks. We set up $Z = 400$ devices in a federated network and distribute the data points among the devices such that each device z only receives data from a random subset of $2 \leq L^{(z)} \leq 4$ clusters with $N^{(z)} \leq 400$ for each $z \in [Z]$. Note that k-means based FC k-FED conventionally underperforms for high-dimensional data, we project the local high-dimensional data to dimensions 10 and 100 by PCA, denoted as PCA-10 and PCA-100, respectively. The performance of evaluated methods on EMNIST and augmented COIL100 is reported in Table III.

Effectiveness of Fed-SC. We can observe the leading clustering performance and the improvement on the connectivity of the affinity graph of Fed-SC (SSC) and Fed-SC (TSC). Besides, the performance of k-FED and that with PCA is worse than our methods. This is because the dimension of local data residing on each device is much higher than the number of data points. The k-means based methods and dimensionality reduction techniques are invalid for high-dimensional data. Furthermore, the proposed Fed-SC significantly improves the clustering accuracy by roughly 30% and 20% over the centralized SC methods on EMNIST and augmented COIL100, respectively. The federated scheme can even boost the performance of central SC methods because Fed-SC can effectively utilize the benefits of heterogeneity. Specifically, due to each device containing data only from a very small set of clusters, it would be much easier to correctly cluster the local data for each device. To verify this insight, we further empirically explore the benefits of statistical heterogeneity for federated clustering on real-world datasets.

We distribute data among Z devices such that each device receives data from a random subset of L' clusters. By varying $L' \in \{2, 4, 6, 8, 10\}$, the clustering performance of the evaluated FC methods are listed in Table IV. We can observe a gradual performance degradation as L' increases and the performance of Fed-SC is even worse than centralized SC methods when $L' = 10$. This verifies that statistical heterogeneity can substantially benefit federated clustering.

Efficiency of Fed-SC. Table III shows that even the most efficient centralized algorithms, TSC and SSCOMP, require computation time about dozens times of that of Fed-SC. Although Fed-SC takes more running time than k-FED, it is still an efficient federated SC scheme, especially when the number of devices is large. This qualifies Fed-SC for large-scale real-world datasets.

VII. CONCLUSION

In this work, we investigated federated clustering for high-dimensional data and proposed the solution of one-shot federated subspace clustering, namely Fed-SC. Fed-SC first enables the effective clustering of high-dimensional data in a federated regime. The effectiveness of Fed-SC is theoretically guaranteed especially with the benefit of statistical heterogeneity. Extensive experiments have been conducted to verify its effectiveness and efficiency. The promising future directions are to theoretically guarantee privacy-preserving and to consider privacy-utility tradeoffs in federated clustering.

REFERENCES

[1] D. K. Dennis, T. Li, and V. Smith, "Heterogeneity for the win: One-shot federated clustering," in *International Conference on Machine Learning*. PMLR, 2021, pp. 2611–2620.

[2] J. Chung, K. Lee, and K. Ramchandran, "Federated unsupervised clustering with generative models," in *International Workshop on Trustable, Verifiable and Auditable Federated Learning*, 2022.

[3] B. van Berlo, A. Saeed, and T. Ozcelebi, "Towards federated unsupervised representation learning," in *ACM International Workshop on Edge Systems, Analytics and Networking*, 2020, pp. 31–36.

[4] F. Zhang, K. Kuang, Z. You, T. Shen, J. Xiao, Y. Zhang, C. Wu, Y. Zhuang, and X. Li, "Federated unsupervised representation learning," *arXiv preprint arXiv:2010.08982*, 2020.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *International Conference on Artificial Intelligence and Statistics*, vol. 54, 2017, pp. 1273–1282.

[6] M. G. Blum, L. Valeri, O. François, S. Cadiou, V. Siroux, J. Lepeule, and R. Slama, "Challenges raised by mediation analysis in a high-dimension setting," *Environmental Health Perspectives*, vol. 128, no. 5, p. 055001, 2020.

[7] K. T. Le, C. Chaux, F. J. Richard, and E. Guedj, "An adapted linear discriminant analysis with variable selection for the classification in high-dimension, and an application to medical data," *Computational Statistics & Data Analysis*, vol. 152, p. 107031, 2020.

[8] R. Vento-Tormo, M. Efremova, R. A. Botting, M. Y. Turco, M. Vento-Tormo, K. B. Meyer, J.-E. Park, E. Stephenson, K. Polański, A. Goncalves *et al.*, "Single-cell reconstruction of the early maternal–fetal interface in humans," *Nature*, vol. 563, no. 7731, pp. 347–353, 2018.

[9] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[10] R. Heckel and H. Bölcskei, "Robust subspace clustering via thresholding," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6320–6342, 2015.

[11] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 1, pp. 1–58, 2009.

[12] C. Bouveyron, S. Girard, and C. Schmid, "High-dimensional data clustering," *Computational Statistics and Data Analysis*, vol. 1, no. 52, pp. 502–519, 2007.

[13] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.

[14] S. R. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[15] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *European Conference on Computer Vision*, 2012, pp. 347–360.

[16] B. McWilliams and G. Montana, "Subspace clustering of high-dimensional data: a predictive approach," *Data Mining and Knowledge Discovery*, vol. 28, no. 3, pp. 736–772, 2014.

[17] M. Soltanolkotabi and E. J. Candes, "A geometric analysis of subspace clustering with outliers," *The Annals of Statistics*, vol. 40, no. 4, pp. 2195–2238, 2012.

[18] Y. Zhao, J. X. Yu, G. Wang, L. Chen, B. Wang, and G. Yu, "Maximal subspace coregulated gene clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 83–98, 2007.

[19] J. Yan and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate," in *European Conference on Computer Vision*, 2006, pp. 94–106.

[20] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid linear modeling via local best-fit flats," *International journal of computer vision*, vol. 100, no. 3, pp. 217–240, 2012.

[21] G. Chen and G. Lerman, "Spectral curvature clustering (scc)," *International Journal of Computer Vision*, vol. 81, no. 3, pp. 317–330, 2009.

[22] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in *IEEE International Conference on Computer Vision*, 2011, pp. 1615–1622.

[23] G. Cormode, S. Muthukrishnan, and W. Zhuang, "Conquering the divide: Continuous clustering of distributed data streams," in *International Conference on Data Engineering*, 2007, pp. 1036–1045.

[24] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[25] C. You, C. Li, D. P. Robinson, and R. Vidal, "A scalable exemplar-based subspace clustering algorithm for class-imbalanced data," in *European Conference on Computer Vision*, 2018, pp. 68–85.

[26] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, "Oracle based active set algorithm for scalable elastic net subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3928–3937.

[27] D. Park, C. Caramanis, and S. Sanghavi, "Greedy subspace clustering," *Advances in Neural Information Processing Systems*, vol. 27, pp. 2753–2761, 2014.

[28] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *International Conference on Knowledge Discovery and Data Mining*, vol. 96, no. 34, 1996, pp. 226–231.

[29] G. Ji and X. Ling, "Ensemble learning based distributed clustering," *Emerging Technologies in Knowledge Discovery and Data Mining*, p. 312, 2007.

[30] N. K. Visalakshi and K. Thangavel, "Distributed data clustering: A comparative analysis," in *Foundations of Computational, IntelligenceVolume 6*. Springer, 2009, pp. 371–397.

[31] M.-F. F. Balcan, S. Ehrlich, and Y. Liang, "Distributed k-means and k-median clustering on general topologies," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[32] E. Januzaj, H.-P. Kriegel, and M. Pfeifle, "Dbdc: Density based distributed clustering," in *International Conference on Extending Database Technology*. Springer, 2004, pp. 88–105.

[33] A. Talwalkar, L. Mackey, Y. Mu, S.-F. Chang, and M. I. Jordan, "Distributed low-rank subspace segmentation," in *IEEE International Conference on Computer Vision*, 2013, pp. 3543–3550.

[34] B. Liu, X.-T. Yuan, Y. Yu, Q. Liu, and D. N. Metaxas, "Parallel sparse subspace clustering via joint sample and parameter blockwise partition," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 3, pp. 1–17, 2017.

[35] J. Li, Z. Tao, Y. Wu, B. Zhong, and Y. Fu, "Large-scale subspace clustering by independent distributed and parallel coding," *IEEE Transactions on Cybernetics*, 2021.

[36] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *International Conference on Data Engineering*, 2022, pp. 965–978.

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

[38] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[39] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (Csur)*, vol. 51, no. 4, pp. 1–35, 2018.

[40] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[41] S. Salehkaleybar, A. Sharifnassab, 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.

[42] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3918–3927.

[43] B. Nasihatkon and R. Hartley, "Graph connectivity in sparse subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2137–2144.

[44] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (gpca)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1945–1959, 2005.

[45] Y. Wang, Y.-X. Wang, and A. Singh, "Graph connectivity in noisy sparse subspace clustering," in *Artificial Intelligence and Statistics*. PMLR, 2016, pp. 538–546.

[46] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: Extending mnist to handwritten letters," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2921–2926.
[47] S. NENE, "Columbia object image library (coil-100)," *Technical Report CUCS-006-96*, 1996.
[48] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.
[49] J. Mairal, R. Jenatton, F. Bach, and G. R. Obozinski, "Network flow algorithms for structured sparsity," *Advances in Neural Information Processing Systems*, vol. 23, 2010.
[50] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.
[51] T. O. Kvalseth, "Entropy and correlation: Some comments," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 17, no. 3, pp. 517–519, 1987.
[52] O. R. Oellermann and A. J. Schwenk, "The laplacian spectrum of graphs," *Graph Theory, c, Appl*, vol. 2, pp. 871–898, 1991.