Fast Algorithms for Distributed k-Clustering with Outliers

Junyu Huang [1 2] **Qilong Feng** [1 2] **Ziyun Huang** [3] **Jinhui Xu** [4] **Jianxin Wang** [1 2 5]

Abstract

In this paper, we study the k-clustering problems with outliers in distributed setting. The current best results for the distributed k-center problem with outliers have quadratic local running time with communication cost dependent on the aspect ratio Δ of the given instance, which may constraint the scalability of the algorithms for handling large-scale datasets. To achieve better communication cost for the problem with faster local running time, we propose an inliers-recalling sampling method, which avoids guessing the optimal radius of the given instance, and can achieve a 4-round bi-criteria $(14(1+\epsilon), 1+\epsilon)$-approximation with linear local running time in the data size and communication cost independent of the aspect ratio. To obtain a more practical algorithm for the problem, we propose another space-narrowing sampling method, which automatically adjusts the sample size to adapt to different outliers distributions on each machine, and can achieve a 2-round bi-criteria $(14(1+\epsilon), 1+\epsilon)$-approximation with communication cost independent of the number of outliers. We show that, if the data points are randomly partitioned across machines, our proposed sampling-based methods can be extended to the k-median/means problems with outliers, and can achieve $(O(\frac{1}{\epsilon^2}), 1+\epsilon)$-approximation with communication cost independent of the number of outliers. Empirical experiments suggest that the proposed 2-round distributed algorithms outperform other state-of-the-art algorithms.

1. Introduction

Clustering is a fundamental problem that has been extensively studied in the past few decades. Different types of clustering problems have been widely used in applications. Since clustering is well known to be sensitive to outliers, one of the major challenges for clustering is how to deal with data noises. Charikar et al. (2001) formulated and studied the clustering problem with outliers, in which a given number z of data points could be discarded as outliers when trying to minimize the clustering cost. For the k-median/means problems with outliers, Chawla and Gionis (2013) modified the Lloyd's algorithm (Lloyd, 1982) to iteratively label the furthest z data points as outliers and adjust the centers according to the labels. By randomized rounding techniques, the k-median/means problems with outliers can be approximated up to a constant factor (Chen, 2008; Krishnaswamy et al., 2018). However, these algorithms have polynomial running time. In order to obtain fast approximation, lots of bi-criteria approximation algorithms have been proposed (Charikar et al., 2001; Zhang et al., 2021; Gupta et al., 2017; Bhaskara et al., 2019; Deshpande et al., 2020; Im et al., 2020). As for the k-center problem with outliers, Charikar et al. (2001) gave a 3-approximation algorithm with running time $O(n^2k)$. Ding et al. (2019) proposed a 2-approximation algorithm with running time $O(\frac{nk}{\epsilon})$ using sampling-based method, which opens $O(k)$ centers and discards $(1+\epsilon)z$ outliers. Recently, Chakrabarty et al. (2016) proposed a reduction-based method that achieves a 2-approximation in polynomial time.

Clustering in distributed setting has also attracted much attention in recent years (Chen et al., 2018; Ding et al., 2016; Dandolo et al., 2022; Guha et al., 2017; Malkomes et al., 2015; Li & Guo, 2018). In this setting, the data points are partitioned among m machines, and a subset $P_i \subseteq P$ of data points with $|P_i| = n_i$ is assigned to machine i. Communications happen in rounds between the coordinator and the machines, where the communication cost is defined as the total number of words sent by the coordinator and machines. In this paper, we study the k-center/median/means problems with outliers in distributed setting, called the distributed (k, z)-center/median/means problems. For the distributed (k, z)-center problem, Malkomes et al. (2015) proposed a $(13+\epsilon)$-approximation algorithm with communication cost $O(m(k+z))$, where the local running time is $O(n_i(k+z))$

[1] School of Computer Science and Engineering, Central South University, Changsha 410083, China [2] Xiangjiang Laboratory, Changsha 410205, China [3] Department of Computer Science and Software Engineering, Penn State Erie, the Behrend College [4] Department of Computer Science and Engineering, State University of New York at Buffalo, NY, USA [5] The Hunan Provincial Key Lab of Bioinformatics, Central South University, Changsha 410083, China. Correspondence to: Qilong Feng <csufeng@mail.csu.edu.cn>, Jianxin Wang <jxwang@mail.csu.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

by picking $(k+z)$ centers on each machine using greedy method. Awasthi et al. (2019) showed that $\Omega(mk+z)$ is the lower bound of communication cost for constant approximation. Guha et al. (2017) proposed an $O(1)$-approximation algorithm by discarding $(2+\delta)z$ outliers with communication cost $\tilde{O}(\frac{m}{\delta}+mk)$, where the local running time is $\tilde{O}(n_i^2)$. Li and Guo (2018) further improved the number of discarded outliers from $(2+\delta)z$ to $(1+\epsilon)z$, and gave a $24(1+\epsilon)$-approximation algorithm with communication cost $O(\frac{km}{\epsilon}\cdot\frac{\log\Delta}{\epsilon})$, where Δ is the aspect ratio of the given instance (aspect ratio of the given instance is defined as the maximum pairwise distance between the given data points divided by the minimum pairwise distance). Their algorithm iteratively removes data points covered by balls with radius $4L^*$, where L^* is the optimal clustering radius (from guessing). It was pointed out in (Li & Guo, 2018) that the quadratic running time on each machine might be the bottleneck for further improvements. Furthermore, since the optimal clustering radius is unknown, enumerations for optimal clustering radius result in a factor of $O(\frac{\log\Delta}{\epsilon})$ on the communication cost and local running time. In (Grunau & Rozhoň, 2022; Bhaskara et al., 2019; Im et al., 2020; Zhang et al., 2021), Δ is assumed to be polynomially bounded. In (Cohen-Addad et al., 2022), a more general case was considered where $\Delta=2^{n^{o(1)}}$, which is much larger than $poly(n)$.

For the distributed (k,z)-median/means problems, Guha et al. (2017) gave a 2-round $O(1)$-approximation algorithm with $(2+\delta)z$ outliers discarded. The communication cost of their algorithm is $\tilde{O}(\frac{m}{\epsilon}+mk)$ with quadratic local running time. Li and Guo (2018) gave a 2-round $(1+\epsilon)$-approximation algorithm by discarding $(1+\epsilon)z$ outliers with communication cost $O(\Phi\cdot\frac{\log(\frac{n\Delta}{\epsilon})}{\epsilon})$, where $\Phi=O(\frac{1}{\epsilon^4}k+mk\log\frac{mk}{\delta})$ and δ is a parameter to control the success probability. However, the running time of their algorithm on the coordinator is exponential. Grunau and Rozhoň (2022) gave a 2-round sampling-based $O(1)$-approximation algorithm by discarding $(1+\epsilon)z$ outliers. The communication cost of their algorithm is $O(\frac{mk\log\Delta}{\epsilon})$ with near-linear local running time in the data size if Δ is assumed to be polynomially bounded, which is the current best result. Although the algorithm given in (Grunau & Rozhoň, 2022) achieves good theoretical guarantee on both communication cost and local running time, the dependence on aspect ratio in communication cost and running time may deteriorate the performance of the algorithm when the aspect ratio is arbitrarily large.

Previously, several sampling-based techniques, such as uniform sampling and D^2-sampling, have been applied to solve the k-clustering problems with outliers. For D^2-sampling methods (Grunau & Rozhoň, 2022; Bhaskara et al., 2019), a guess for the optimal clustering cost is needed to serve as a threshold for distance penalization, which leads to a dependence on the aspect ratio in communication cost and local running time. Ding et al. (2019) proposed a uniform sampling algorithm for the k-center problem with outliers. It is required that the exact number of outliers to be discarded should be given. However, in distributed setting, the exact number of outliers on each machine is unknown.

1.1. Our Contribution

To improve the running time on each machine and avoid the dependence on the aspect ratio in communication cost, we propose a sampling-based method (called inliers-recalling sampling) for the distributed (k,z)-center problem. On each machine, $(1+\epsilon)z$ data points are discarded using uniform sampling methods in order to remove the aspect ratio Δ from the communication cost and guarantee an approximation ratio of 2. Then, the inliers-recalling sampling procedure is used to recover as many inliers as possible.

Theorem 1.1. *For the distributed (k,z)-center problem, there exists a 4-round distributed algorithm that outputs a $(14(1+\epsilon),1+\epsilon)$-approximate solution with constant probability, where the communication cost is $O(\frac{m^3k\log(mk)}{\epsilon^2})$, and the running time on each machine i is $O(\frac{n_im^3k\log(mk)}{\epsilon^2})$.*

Inliers-recalling sampling takes four rounds of communication (between each machine and the coordinator) to avoid the dependence on Δ in the communication cost. This may impact its practical performance, and may not be ideal for the case when Δ is small. To avoid this issue and obtain a more practical algorithm, we propose another sampling-based method, called space-narrowing sampling, which adaptively adjusts the sample size to avoid the requirement that the exact number of outliers should be given in the local computations. With this technique, the total number of discarded inliers on all machines can be bounded by ϵz.

Theorem 1.2. *For the distributed (k,z)-center problem, there exists a 2-round distributed algorithm that outputs a $(14(1+\epsilon),1+\epsilon)$-approximate solution with constant probability, where the communication cost is $O(\frac{mk\log m}{\epsilon}\cdot\frac{\log\Delta}{\epsilon})$, and the local running time on each machine i is $O(\frac{n_ik\log m}{\epsilon}\cdot\frac{\log\Delta}{\epsilon})$.*

For the distributed (k,z)-median/means problems, if the data points are randomly partitioned across the machines, we show that by combining our proposed sampling-based methods with a filtering process, we can obtain a $(O(\frac{1}{\epsilon^2}),1+\epsilon)$-approximation algorithm with communication cost $O(\frac{mk\log m\log n}{\epsilon}\cdot\frac{\log\Delta}{\epsilon})$ and local running time $O(\frac{n_ik\log m\max\{\log n_i,k\}}{\epsilon}\cdot\frac{\log\Delta}{\epsilon})$ using the space-narrowing sampling method. By combining our proposed inliers-recalling sampling method with the filtering process, we can obtain a $(O(\frac{1}{\epsilon^2}),1+\epsilon)$-approximate solution with com-

Table 1. Comparison results for distributed (k, z)-center

Approximation	Communication	Local Running Time	Reference
$13(1+\epsilon)$	$O(m(k+z))$	$O(n_i(k+z))$	(Charikar et al., 2001)
$(O(1), 2+\epsilon)$	$O((\frac{m}{\epsilon}+mk)\log n)$	$O(n_i^2 \log n)$	(Guha et al., 2017)
$(24(1+\epsilon), 1+\epsilon)$	$O(\frac{mk}{\epsilon} \cdot \frac{\log \Delta}{\epsilon})$	$O(\frac{n_i^2 \log \Delta}{\epsilon})$	(Li & Guo, 2018)
$(14(1+\epsilon), (1+\epsilon))$	$O(\frac{km^3 \log(mk)}{\epsilon^2})$	$O(\frac{n_i m^3 k \log(mk)}{\epsilon^2})$	Inliers-Recalling Sampling
$(14(1+\epsilon), (1+\epsilon))$	$O(\frac{mk \log m}{\epsilon} \cdot \frac{\log \Delta}{\epsilon})$	$O(\frac{n_i k \log m}{\epsilon} \cdot \frac{\log \Delta}{\epsilon})$	Space-Narrowing Sampling

Table 2. Comparison results for distributed (k, z)-median/means

Approximation	Communication	Local Running Time	Reference
$(1+\epsilon, 1+\epsilon)$	$O((\frac{k}{\epsilon^4}+mk\log\frac{mk}{\delta}) \cdot \frac{\log \frac{n\Delta}{\epsilon}}{\epsilon})$	Polynomial	(Li & Guo, 2018)
$(O(\frac{1}{\epsilon}), 2+\epsilon+\delta)$	$O((\frac{m}{\epsilon}+mk)\log n)$	$O(n_i^2)$	(Guha et al., 2017)
$O(1)$	$O((k\log n+z)m)$	$O(n_i \max\{k, \log n\})$	(Chen et al., 2018)
$(O(1), 1+\epsilon)$	$O(\frac{mk \log \Delta}{\epsilon})$	$O(\frac{n_i k \log \Delta}{\epsilon})$	(Grunau & Rozhoň, 2022)
$O(\frac{1}{\epsilon^2}, 1+\epsilon)$	$O(\frac{m^3 k \log n \log(mk)}{\epsilon^2})$	$O(\frac{n_i m^3 k \log(mk) \max\{\log n_i, k\}}{\epsilon^2})$	Inliers-Recalling Sampling
$O(\frac{1}{\epsilon^2}, 1+\epsilon)$	$O(\frac{mk \log m \log n}{\epsilon} \cdot \frac{\log \Delta}{\epsilon})$	$O(\frac{n_i k \log m \max\{\log n_i, k\}}{\epsilon} \cdot \frac{\log \Delta}{\epsilon})$	Space-Narrowing Sampling

munication cost $O(\frac{m^3 k \log n \log(mk)}{\epsilon^2})$ and local running time $O(\frac{n_i m^3 k \log(mk) \max\{\log n_i, k\}}{\epsilon^2})$.

Table 1 and Table 2 show the detailed comparison results for the distributed (k, z)-center/median/means problems. It can be seen that, compared with other distributed (k, z)-center algorithms, our proposed inliers-recalling sampling method has linear local running time in the data size, where the communication cost is independent of Δ with better approximation guarantee on the clustering cost.

2. Preliminaries

Let P be the given set of data points with size n in metric space $(\mathcal{X}, d)$. For two points p, $q \in P$, let $d(p, q)$ denote their distance. For any point $p \in P$ and a subset $C \subseteq P$, let $d(p, C) = \min_{c \in C} d(p, c)$. For any two subsets $C, C' \subseteq P$, let $d(C, C') = \min_{c \in C} d(c, C')$ and $\delta(C, C') = \max_{c \in C} d(c, C')$, respectively. Given the number of clusters k and the number of outliers z, the goal of the (k, z)-center problem is to find a set $C \subseteq P$ of k centers and a set $Z \subseteq P$ of z outliers such that clustering cost $\max_{p \in P \setminus Z} d(p, C)$ is minimized. For a fixed set C of centers, the outliers discarded can be obtained by choosing the furthest z points from C. We use C^* to denote an optimal set of centers. Based on C^*, let Z^* be the set of the furthest z outliers from C^*. Let $P_{opt} = P \setminus Z^*$ be the set of inliers, and let $L^* = \max_{p \in P_{opt}} d(p, C^*)$ be an optimal radius. Based on C^*, by discarding the points in Z^*, we can find k optimal clusters, denoted by $D^* = \{D_1^*, \ldots, D_k^*\}$. Let $[m] = \{1, \ldots, m\}$. For any point $p \in P$ and a radius r, the set of points in P covered by p with radius r is denoted as $B_P(p, r)$. For a subset $Q \subseteq P$, define $B_P(Q, r) = \cup_{q \in Q} B_P(q, r)$. For two subsets C, $D \subseteq P$, if each point in C is covered by at least one point in D with radius r, then it is called that C is covered by the points in D with radius r. For a random event E, let $P_r(E)$ denote the probability that event E occurs. We say that an algorithm for the (k, z)-center/median/means problems achieves a bi-criteria approximation ((α, β)-approximation) for some α, $\beta \geq 1$, if it outputs a solution with at most βz outliers, whose cost is at most α times the cost of the optimum solution.

Computation Model. We follow the same distributed setting as in (Li & Guo, 2018), where all communications need to go through a central coordinator. Following the one in (Li & Guo, 2018), our distributed algorithms only need to output a set C of k centers on the coordinator as the solution.

3. Algorithm for Distributed (k, z)-Center without Δ on Communication Cost

For the distributed (k, z)-center problem, the number of inliers discarded by m machines should be less than ϵz before executing a central clustering algorithm on the coordinator. Thus, a key challenge is to ensure that the sampling method discards no more than ϵz inliers across m machines.

In this section, we give an inliers-recalling sampling method to remove Δ from the communication cost. Our proposed method is based on a 2-approximate solution with $(1+\epsilon)z$ data points discarded as an initialization on each machine. To reduce the number of inliers discarded, we propose a sampling process with two stages to recall back as many inliers as possible. In the first stage, by randomly taking a small sample from the $(1+\epsilon)z$ discarded data points and adding them to the candidate set of centers, there are at most $\frac{\epsilon z}{2m}$ inliers with distances larger than $2L^*$ to the candidate set of centers, where L^* is an optimal clustering radius of the given instance. In the second stage, on each machine, the algorithm iteratively removes the nearest $\frac{\epsilon z}{2m}$ data points from the discarded data points. By repeating the process for $O(\frac{m}{\epsilon})$ rounds, a list of clustering radii with size $O(\frac{m}{\epsilon})$ can be obtained. Moreover, we can get that there exists at least one radius in the list such that at most $\frac{\epsilon z}{m}$ inliers are discarded on each machine, which ensures that the total number of inliers discarded across the machines can be bounded by ϵz. In the following, subscript i represents the i-th machine, and subscript j represents the j-th iteration of the "for" loop.

Assume that we execute the inliers-recalling sampling (Algorithm 1) on machine i ($i \in [m]$) with a set $P_i \subseteq P$ of data points. Let S_i be the set of data points sampled in step 3 of Round 1 in Algorithm 1. We first prove that, with constant probability, $|S_i \cap P_{opt}| > 0$. Due to space limit, all the proofs in section 3 are given in Appendix A.

Lemma 3.1. *By executing Algorithm 1 on machine i, with probability at least* $1-\eta$*, the set* S_i *sampled in step* 3 *of Round 1 in Algorithm 1 contains at least one point from* $P_{opt} \cap P_i$.

There are two cases that may happen during the sampling process in steps 4-7 of Round 1 in Algorithm 1: (1) a 2-approximate solution can be obtained by discarding the furthest $(1+\epsilon)z$ data points; (2) at least one optimal cluster can be covered by the data points sampled in step 6 of Round 1 in Algorithm 1. In the following, we discuss the two cases separately. In the j-th iteration of the for loop in step 4 of Round 1 in Algorithm 1, let $Q_{i,j}$ be the candidate set of centers obtained in step 6 of Round 1, and let $R_{i,j}$ be the set of the furthest $(1+\epsilon)z$ points from $Q_{i,j-1}$ in step 5 of Round 1. Let $\beta_i(Q_{i,j}) \subseteq [k]$ denote the set of indices of optimal clusters covered by the points in $Q_{i,j}$ with radius $2L^*$, i.e., $h \in \beta_i(Q_{i,j})$ if $\delta(D_h^* \cap P_i, Q_{i,j}) \leq 2L^*$.

Lemma 3.2. *In each iteration* j *of the for loop in step 4 of Round 1 in Algorithm 1 on machine i, either* $d(R_{i,j}, Q_{i,j-1}) \leq 2L^*$ *or* $|\beta_i(Q_{i,j})| > |\beta_i(Q_{i,j-1})|$ *with probability at least* $1-\eta$.

We first argue that a 2-approximate solution can be obtained if $d(R_{i,j}, Q_{i,j-1}) \leq 2L^*$. Note that $R_{i,j}$ is the set of the furthest $(1+\epsilon)z$ points from $Q_{i,j-1}$. Thus, for each point $p \in P_i \backslash R_{i,j}$, we have $d(p, Q_{i,j-1}) \leq d(R_{i,j}, Q_{i,j-1}) \leq 2L^*$. Hence, a 2-approximate solution can be obtained by discarding the data points in $R_{i,j}$. Then, consider the case that $d(R_{i,j}, Q_{i,j-1}) > 2L^*$ holds for each $j \in [\lambda]$ during the sampling process in steps 4-7, where $\lambda = \gamma k \log m$ and γ is a large constant. Let $Q_{i,\lambda}$ be the candidate set of centers obtained after λ rounds. We prove that with probability at least $1-\frac{\eta}{m}$, points in $\bigcup_{t=1}^{k} D_t^* \cap P_i$ are all covered by the points in $Q_{i,\lambda}$ with radius $2L^*$.

Algorithm 1 IRS

```
Input: A partition $(P_1, P_2, ..., P_m)$ of dataset $P$ among $m$ machines, and parameters $k$, $z$, $\eta$, $\epsilon$.
Output: A collection of weighted representations.
Round 1 on each machine $i \in [m]$
 1: Initialize $Q_i = \emptyset$, and $L_i = \emptyset$.
 2: if $|P_i| > (1+\epsilon)z$ then
 3:     Randomly sample a subset $S_i \subseteq P_i$ of size $\frac{1+\epsilon}{\epsilon}\log\frac{1}{\eta}$, and add data points in $S_i$ to $Q_i$.
 4:     for $j = 1, 2, \ldots, O(k\log m)$ do
 5:         Let $R_j \subseteq P_i$ be the set of the furthest $(1+\epsilon)z$ points to $Q_i$.
 6:         Randomly sample a subset $S_j \subseteq R_j$ of size $\frac{1+\epsilon}{\epsilon}\log\frac{1}{\eta}$, and add data points in $S_j$ to $Q_i$.
 7:     end for
 8: end if
 9: Let $R_i \subseteq P_i$ be the set of the furthest $(1+\epsilon)z$ data points from $Q_i$, and set $L_i = L_i \cup \{\delta(P_i \backslash R_i, Q_i)\}$.
10: Randomly sample a subset $V_i \subseteq R_i$ of size $\frac{2km(1+\epsilon)}{\epsilon}\log\frac{km}{\eta}$, and add data points in $V_i$ to $Q_i$.
11: for $j = 1$ to $\lceil\frac{2m(1+\epsilon)}{\epsilon}\rceil$ do
12:     Let $T_j \subseteq R_i$ be the set of the nearest $\frac{\epsilon z}{2m}$ data points from $Q_i$.
13:     $R_i = R_i \backslash T_j$.
14:     $l_j = \delta(P_i \backslash R_i, Q_i)$, and $L_i = L_i \cup \{l_j\}$.
15: end for
16: Send $L_i$ to the coordinator.
Round 2 on coordinator
 1: Let $L' = \bigcup_{i\in m} L_i$ be the collection of radii.
 2: Send $L'$ to each machine $i \in [m]$.
Round 3 on each machine $i \in [m]$
 1: for $L \in L'$ do
 2:     $S_i^L = Q_i$, $H_s = \cup_{s_i \in S_i^L} B_{P_i}(s_i, L)$.
 3:     Assign each point in $H_s$ to its closest center in $S_i^L$.
 4:     For each $s \in S_i^L$, let $\sigma(s)$ be the number of points assigned to $s$, and assign a weight $\sigma(s)$ to $s$.
 5:     Send $(S_i^L, L)$ to the coordinator.
 6: end for
```

Lemma 3.3. ***Chernoff Bound .*** *Let* $x_1, x_2, ..., x_n$ *be* n *independent random variables with values* 1 *or* 0*, where* x_i *takes value* 1 *with probability at least* q *for each* $i = 1, 2, ..., n$*. Let* $X = \sum_{i=1}^{n} x_i$*. For any real number* $\epsilon \in (0,1]$*, we have* $P_r\left(X < (1-\epsilon)E(X)\right) < e^{-\frac{\epsilon^2 qn}{2}}$.

Lemma 3.4. *For each machine* $i \in [m]$*, assume that* $d(R_{i,j}, Q_{i,j-1}) > 2L^*$ *holds for each* $j \in [\lambda]$ *during the sampling process in steps 4-7 of Round 1 in Algorithm 1,*

Algorithm 2 MCA

```
Input: a set $P'$ of weighted representations, and parameters $z'$, $R$.
Output: A set $C' \subseteq P$ of centers with size at most $k$ and a radius $r$.
Let $d_{max}$ be the maximum pairwise distance in $P'$, $l = \lfloor \log_{1+\epsilon} \frac{R}{2} \rfloor$, $u = \lceil \log_{1+\epsilon} d_{max} \rceil$. For each point $p \in P'$, let $w(p)$ be the weight of $p$.
for $i = l$ to $u$ do
    $L = (1+\epsilon)^i$, $Y = P'$ and $C' = \emptyset$.
    for $j = 1$ to $k$ do
        $c_j = \arg\max_{p \in Y} \sum_{q \in B_Y(p, 6L)} w(q)$.
        $C' = C' \cup \{c_j\}$, and $Y = Y \backslash B_Y(c_j, 12L)$.
    end for
    if $\sum_{p \in Y} w(p) \leq z'$ then
        Return $(C', L)$.
    end if
end for
```

Algorithm 3 Distributed (k, z)-center using IRS

```
Input: A partition $\{P_1, P_2, ..., P_m\}$ of dataset $P$ among $m$ machines, and parameters $k$, $z$, $\eta$, $\epsilon$, $L$.
Output: A set $C \subseteq P$ of centers with size at most $k$.
Rounds 1-3
IRS($\{P_1, P_2, ..., P_m\}, k, z, \eta, \epsilon$).
Round 4 on coordinator
Initialize $C_f = \emptyset$, and $L_f = \infty$.
for $L \in L'$ do
    $P_L = \cup_{i=1}^{m} S_i^L$.
    For each point $p$ in $P_L$, let $w(p)$ be the weight of $p$.
    $z' = (1+\epsilon)z + \sum_{p \in P_L} w(p) - |P|$.
    if $z' > 0$ then
        $(C_t, L_t) = MCA(P_L, z', L)$.
        if $L_t < L_f$ then
            $L_f = L_t, C_f = C_t$.
        end if
    end if
end for
Return $C_f$.
```

where $\lambda = \gamma k \log m$, and γ is a large constant. Then, with probability at least $1 - \frac{\eta}{m}$, $\beta_i(Q_{i,\lambda}) = k$.

By Lemma 3.4, in step 9 of Round 1 in Algorithm 1, if the data points in R_i are discarded as outliers, a set Q_i of candidate centers with approximation ratio 2 on machine i can be obtained. By taking a union bound over the success probability across machines, a 2-approximate solution can be obtained on each machine with probability at least $1 - \eta$.

However, in the worst case, the discarded $(1+\epsilon)z$ data points in R_i are all inliers. Then, a sampling process with two stages is used to reduce the number of discarded inliers. In the first stage (step 10 of Round 1 in Algorithm 1), by sampling data points from R_i as new centers, we can guarantee that there are few inliers with distances larger than $2L^*$ to the clustering centers. More formally, a small sample V_i is randomly taken from R_i to guarantee that for any optimal cluster D_t^* with $|D_t^* \cap R_i| \geq \frac{\epsilon z}{2km}$, $|D_t^* \cap V_i| > 0$ happens with certain probability. Observe that if $|D_t^* \cap R_i| \geq \frac{\epsilon z}{2km}$, $\frac{|D_t^* \cap R_i|}{|R_i|} \geq \frac{\epsilon}{2(1+\epsilon)km}$. By Lemma 3.1, if randomly taking a sample V_i of size $\frac{2(1+\epsilon)km}{\epsilon} \log \frac{1}{\eta}$ from R_i, with probability at least $1 - \eta$, one point from $D_t^* \cap R_i$ can be sampled. A union bound on success probability over all clusters and machines can be obtained by replacing η with $\frac{mk}{\eta}$. Then, by adding data points in V_i to Q_i, with probability at least $1 - \frac{\eta}{m}$, the total number of uncovered inliers by data points in Q_i with radius $2L^*$ can be bounded by $\frac{\epsilon z}{2m}$ on each machine i.

In the second stage (steps 11-15 in Round 1 of Algorithm 1), a recalling process is used to find most discarded inliers back such that a unified clustering radius can be obtained across machines. On each machine $i \in [m]$, the goal is to find as many inliers as possible with distances smaller than $2L^*$ to Q_i. There are at most $O(\frac{m(1+\epsilon)}{\epsilon})$ iterations to recall the inliers, in which the nearest $\frac{\epsilon z}{2m}$ data points in R_i to Q_i are iteratively removed from R_i in steps 12-13. It holds trivially that there must exist at least one iteration such that at most $\frac{\epsilon z}{2m}$ inliers in the discarded $(1+\epsilon)z$ outliers with distances smaller than $2L^*$ to Q_i are not recalled back. Hence, in step 16, we can find a radius $L_i^f \in L_i$ such that $L_i^f \leq 2L^*$, and by using data points in Q_i with radius L_i^f, there are at most $\frac{\epsilon z}{m}$ inliers that are not covered ($\frac{\epsilon z}{2m}$ inliers with distances larger than $2L^*$to Q_i, and $\frac{\epsilon z}{2m}$ inliers without being recalled back). Let $L_f = \max_{i \in [m]} L_i^f$. Putting all things together, since $L_f \leq 2L^*$, with probability at least $(1-\eta)^2$, a 2-approximate solution on each machine i can be obtained using data points in Q_i with radius L_f in Round 2 of Algorithm 1, where at most ϵz inliers are discarded across machines.

Corollary 3.5. *In Round 2 of Algorithm 1, with constant probability, there exists at least one clustering radius $L_f \in L'$ such that $L_f \leq 2L^*$, and for each machine $i \in [m]$, $|P_{opt} \cap (P_i \backslash B_{P_i}(Q_i, L_f))| \leq \frac{\epsilon z}{m}$.*

The algorithm solving the distributed (k, z)-center by inliers-recalling sampling is given in Algorithm 3. In the first round, each machine conducts random sampling and inliers recalling to obtain a list of clustering radii, and sends the list of radii to the coordinator. In the second round, the coordinator collects the radii lists from machines, and sends the collection of radii back to each machine. In the third round, machines send back all the weighted representations constructed using the radii list sent by the coordinator. In the fourth round, the coordinator performs a weighted clustering algorithm to obtain a final solution. Note that the radii list on each machine has size at most $O(\frac{m}{\epsilon})$. Thus, the total number of possible radii can be bounded by $O(\frac{m^2}{\epsilon})$. On each machine, there are at most $O(\frac{mk\log(mk)}{\epsilon})$ centers sampled as representations. Then, the total communication cost is $O(\frac{m^3k\log(mk)}{\epsilon^2})$.

Corollary 3.6. *The total communication cost of Algorithm 1 is $O(\frac{m^3k\log(mk)}{\epsilon^2})$.*

The algorithm solving the weighted (k,z)-center problem is given in Algorithm 2. We first show how to obtain a bound for L^*. By Corollary 3.5, there exists a radius $L_f \in L'$ such that $L_f \leq 2L^*$ and the number of uncovered inliers across machines is at most ϵz. Let P_{L_f} denote the collection of weighted representations with radius L_f in step 3 of Round 4 in Algorithm 3, and let d_{max} be the maximum pairwise distance in P_{L_f}. Since $L_f \leq 2L^*$, L^* is in range $[\frac{L_f}{2}, d_{max}]$.

The goal of Algorithm 2 is to greedily find a point that covers the most data points within distances $6L^*$. We show that such a greedy strategy works. In step 2 of Algorithm 2, assume that an estimation L of L^* with $L \in [L^*, (1+\epsilon)L^*]$ is obtained by enumeration. Let H be the set of the data points discarded across machines by P_{L_f}. Let $X = P \backslash H$ be the set of the covered data points. For each $p \in X$, we have $d(p, P_{L_f}) \leq L_f \leq 2L^*$ by Corollary 3.5. In step 4 of Algorithm 2, there are k iterations. In the j-th iteration, let Y_j be the set of the uncovered data points before adding a center c_j to C' in step 6. For any point $p \in P_i$, if p is covered by a point $q \in Q_i$ with radius $2L^*$, then denote $s_p = q$ as the representation of p. In the j-th iteration of Algorithm 2, for an optimal cluster D_h^*, let $U_{D_h^*} = \{p \in D_h^* \cap X : s_p \in Y_j\}$ be the set of the data points in D_h^* whose representations are not covered before the j-th iteration.

Lemma 3.7. *In the j-th iteration of Algorithm 2, let c_j be the center added to C' in step 6. Then, for each uncovered optimal cluster D_h^*, $\sum_{p \in B_{Y_j}(c_j, 6L)} w(p) \geq |U_{D_h^*}|$.*

Lemma 3.8. *In the j-th iteration of Algorithm 2, let $\lambda_j = B_{Y_j}(c_j, 12L)$. Then, $\sum_{j=1}^{k} \sum_{p \in \lambda_j} w(p) \geq |P_{opt} \cap X|$.*

By Lemma 3.8, it holds that the number of covered inliers on the coordinator is at least the number of covered inliers across machines. Together with Corollary 3.5, the number of data points discarded can be bounded by $(1+\epsilon)z$.

Corollary 3.9. *In Algorithm 3, the number of data points with distances large than $2L^*$ to C_f is bounded by $(1+\epsilon)z$.*

The proof of Theorem 1.1 is in Appendix A.

4. A More Practical Algorithm with Smaller Communication Rounds

Although the aspect ratio for a given instance may be arbitrarily large, experiments show that for many datasets, Δ is actually much smaller and is often a small constant, which makes the removal of the dependence on Δ in the communication cost less exciting. Algorithm 3 needs four rounds of communication between machines and the coordinator, which could deteriorate the practical performance of the algorithm. To obtain a much more practical algorithm for small Δ, we propose another sampling method, called space-narrowing sampling, based on a guess for the optimal clustering radius L^*. The space-narrowing sampling method can adjust the sample size according to the number of uncovered points to avoid the requirement that the exact number of outliers should be given on each machine.

The space-narrowing sampling process is given in Algorithm 4, which is executed on each machine $i \in [m]$. The basic idea behind is to iteratively use random sampling method to find some centers and remove the data points covered by the centers with radius $2L^*$. On each machine i, let $U_{i,j}$ be the set of the uncovered data points before the execution of the j-th iteration of step 2 in Algorithm 4. For a single machine $i \in [m]$, during the sampling process, there are two cases that may happen: (1) there are enough uncovered inliers, i.e., $|U_{i,j}| \geq (1+\epsilon)z$; (2) the number of uncovered data points is small, i.e., $|U_{i,j}| < (1+\epsilon)z$. In the following, we discuss the two cases separately. According to the results of (Li & Guo, 2018), we can obtain an estimation L of optimal clustering radius L^* such that $L \in [L^*, (1+\epsilon)L^*]$ by trying $O(\frac{\log \Delta}{\epsilon})$ radii to serve as the input of Algorithm 5, which results in an $O(\frac{\log \Delta}{\epsilon})$ factor on communication cost and running time.

Assume that we execute Algorithm 4 on machine i ($i \in [m]$). In Algorithm 4, all data points sampled are added to a weighted set Q of centers in step 10. Assume that $Q_{i,j}$ is the set of centers found after the j-th iteration of the for loop in step 2. Let $\beta_i(Q_{i,j}) \subseteq [k]$ denote the set of indices of optimal clusters covered by the points in $Q_{i,j}$, i.e., $h \in \beta_i(Q_{i,j})$, if $\delta(D_h^* \cap P_i, Q_{i,j}) \leq 2L$. By Lemma 3.1, if there are enough uncovered data points left in $P_i \cap P_{opt}$, i.e., $|U_{i,j}| \geq (1+\epsilon)z$, in the j-th iteration, we can sample at least one point $p \in P_{opt} \cap U_{i,j}$ with constant probability in step 7. Let D_t^* be the optimal cluster containing p. By removing the data points covered by p with radius $2L$ in P_i, since $L \in [L^*, (1+\epsilon)L^*]$, all the points in $D_t^* \cap P_i$ are covered. Thus, $|\beta_i(Q_{i,j})| > |\beta_i(Q_{i,j-1})|$. If case (2) never happens during the sampling process, Lemma 3.4 shows that by repeating the sampling process $O(k \log m)$ times, with probability at least $1 - \frac{\eta}{m}$, the points in $\bigcup_{t=1}^{k} D_t^* \cap P_i$ are all covered by the points in Q_i with L, where Q_i is a set of weighted representations returned by Algorithm 4 on machine i.

For case (2), assume that $|U_{i,j}| < (1+\epsilon)z$, which means that there are not enough uncovered inliers. In this case, we need to carefully adjust the sample size to guarantee that at least one uncovered inlier can still be sampled. We first show that, before each execution of step 7 of Algorithm 4, we can always bound the size of $U_{i,j}$ to guess the number of uncovered inliers. Before step 7, we can find an integer $r_{i,j}$ such that $\frac{(1+\epsilon_1)\epsilon_1 r_{i,j} z}{m} \leq |U_{i,j}| < \frac{(1+\epsilon_1)\epsilon_1 (r_{i,j}+1) z}{m}$, where $\epsilon_1 = \frac{\epsilon}{3}$. Then, there are two subcases to consider. In the first subcase, we have $|Z^* \cap U_{i,j}| < \frac{\epsilon_1 r_{i,j} z}{m}$. In this subcase, we show that the total number of uncovered inliers is roughly bounded by $\frac{\epsilon z}{m}$. In the second subcase, we have $|Z^* \cap U_{i,j}| \geq \frac{\epsilon_1 r_{i,j} z}{m}$, where inliers can still be picked with certain probability by increasing the sample size.

Algorithm 4 SNS

```
Input: A set $P_i$ of data points, and parameters $k$, $z$, $\eta$, $\epsilon$, $m$, $L$.
Output: A weighted set $Q$ of representations.
 1: Initialize $Q = \emptyset$, and $U = P_i$.
 2: for $i = 1, 2, ..., O(k \log m)$ do
 3:     $\epsilon_1 = \epsilon$.
 4:     if $|U| < (1+\epsilon)z$ then
 5:         $\epsilon_1 = \frac{\epsilon}{3}$.
 6:     end if
 7:     Randomly sample a subset $S \subseteq U$ of size $\frac{1+\epsilon_1}{\epsilon_1} \log \frac{1}{\eta}$.
 8:     for each $s \in S$ do
 9:         $H_s = B_U(s, 2L)$.
10:         Assign a weight $|H_s|$ to point $s$, add $s$ to $Q$, and set $U = U \backslash H_s$.
11:     end for
12: end for
13: Return $Q$.
```

In the following, by considering all m machines together, we give a bound on the number of uncovered inliers. We first assume that subcase 2 never happens on each machine, i.e., $|Z^* \cap U_{i,j}| < \frac{\epsilon_1 r_{i,j} z}{m}$ for each $i \in [m]$ and $j \in [\lambda \lceil k \log m \rceil]$, where λ is a large constant. In this subcase, data points in $\bigcup_{t=1}^{k} D_t^* \cap P_i$ ($i \in [m]$) can be covered by the points in Q_i with radius $2L$, where $L \in [L^*, (1+\epsilon)L^*]$. Note that on each machine i ($i \in [m]$), we have $|P_{opt} \cap U_{i,j}| \geq \epsilon_1 |Z^* \cap U_{i,j}|$. Let $S_{i,j}$ be the set of the data points sampled in step 7 of Algorithm 4 in the j-th iteration. Then, by Lemma 3.1, with probability at least $1-\eta$, the set $S_{i,j}$ contains at least one uncovered point from P_{opt}. Thus, by taking a union bound of the success probability over all machines, we can get that with probability at least $1-\eta$, $\beta_i(Q_i)$ is equal to k for each machine $i \in [m]$ by Lemma 3.4, where the total number of uncovered inliers is 0. On the other hand, we consider that subcase 2 happens on a subset $F \subseteq [m]$ of machines. In this case, we show that the total number of uncovered inliers is bounded by ϵz. Due to space limit, proof of Lemma 4.1 is given in Appendix B.

Lemma 4.1. *Let $F \subseteq [m]$ denote the set of machines where subcase 2 happens at least once during the space-narrowing sampling process. Then, the total number of uncovered inliers is bounded by ϵz.*

Algorithm 5 Distributed (k, z)-center using SNS

```
Input: A partition $\{P_1, P_2, ..., P_m\}$ of data set $P$ among $m$ machines, parameters $k$, $z$, $\eta$, $\epsilon$, $L$.
Output: A set $C \subseteq P$ of size at most $k$ or "No".
Round 1 on each machine $i \in [m]$
    $Q_i$ = SNS($P_i$, $k$, $z$, $\eta$, $\epsilon$, $m$, $L$).
    Send the weighted representation $Q_i$ to the coordinator.

Round 2 on coordinator
    $P' = \cup_{i \in [m]} Q_i$.
    For each point $p$ in $P'$, let $w(p)$ be the weight of $p$.
    $z' = (1+\epsilon)z + \sum_{p \in P'} w(p) - |P|$.
    if $z' < 0$ then
        Return "No".
    else
        Return MCA($P'$, $z'$, $L$).
    end if
```

Our main algorithm for the distributed (k, z)-center problem is given in Algorithm 5, which consists of two rounds. In the first round, each machine conducts space-narrowing sampling to obtain a set of weighted representations, and sends the result to the coordinator. In the second round, the coordinator collects the weighted representations, and executes the algorithm (given in Algorithm 2) for the weighted (k, z)-center to obtain the final clustering results.

Corollary 4.2. *The total communication cost of Algorithm 5 is $O(\frac{mk \log m}{\epsilon} \cdot \frac{\log \Delta}{\epsilon})$.*

The proof of Theorem 1.2 is in Appendix B.

5. Extension to the k-Median/Means Problems with Outliers

For the k-means/median problems with outliers, although the uniform sampling method in (Chen et al., 2018) can achieve a communication cost independent of Δ, the communication cost has linear dependence on the number of outliers. For non-uniform sampling methods, the sampling process should rely on a guess for the optimal clustering cost to trim the distances between data points and their centers, which leads to an $O(\log \Delta)$ loss on both communication cost and running time. Thus, it is challenging to obtain an approximation algorithm with communication cost independent of aspect ratio, where the local running time is near-linear in the data size on each machine.

We sketch the high-level idea of the process solving the k-median/means problems, and the detailed algorithms are in Appendix C. The general idea behind is to firstly use a filtering process to obtain a constant approximate solution with $O(z)$ data points discarded on each machine. We can use the algorithm proposed in (Chen et al., 2018) as the filtering process, which is given in Appendix C. According to the random partition rule, we show that the number of discarded data points can be bounded by $O(\frac{z}{m})$ on each machine. By taking a small sample from the discarded data points using our proposed sampling-based methods, at most ϵz inliers are discarded across the machines. Furthermore, the loss on approximation ratio can be bounded by $O(\frac{OPT}{\epsilon})$. Hence, we can get a set of weighted representations with $O(\frac{1}{\epsilon})$-approximation on the coordinator by discarding at most ϵz inliers across the machines. By executing a weighted $(O(\frac{1}{\epsilon}), 1+\epsilon)$-approximation algorithm on the coordinator, such as the LS++ algorithm given in (Grunau & Rozhoň, 2022), we can get a $(O(\frac{1}{\epsilon^2}), 1+\epsilon)$-approximate solution on the coordinator. Putting these together, we have the following results for the distributed (k, z)-median/means problems.

Theorem 5.1. *For the distributed (k, z)-median/means problem, if the data points are randomly partitioned across the machines, there exists a 4-round distributed algorithm that outputs a $(O(\frac{1}{\epsilon^2}), 1+\epsilon)$-approximate solution with constant probability, where the communication cost and the running time on machines are $O(\frac{m^3 k \log n \log(mk)}{\epsilon^2})$ and $O(\frac{n_i m^3 k \log(mk) \max\{\log n_i, k\}}{\epsilon^2})$, respectively.*

Theorem 5.2. *For the distributed (k, z)-median/means problem, if the data points are randomly partitioned across the machines, there exists a 2-round distributed algorithm that outputs a $(O(\frac{1}{\epsilon^2}), 1+\epsilon)$-approximate solution with constant probability, where the communication cost and the running time on machines are $O(\frac{mk \log m \log n}{\epsilon} \cdot \frac{\log \Delta}{\epsilon})$ and $O(\frac{n_i k \log m \max\{\log n_i, k\}}{\epsilon} \cdot \frac{\log \Delta}{\epsilon})$, respectively.*

6. Experiments

In this section, we evaluate the performance of our algorithms on several real-world datasets[1] including 3 small datasets (letter: 20,000 × 16, skin: 245,057 × 3, covertype: 581,012 × 54) and 3 large datasets (gas: 928,991 × 10 and higgs: 11,000,000 × 27, sift[2]: 100,000,000 × 128). The datasets for testing the k-center and k-means algorithms are different in (Li & Guo, 2018). For fair comparison, we use the same datasets as in (Li & Guo, 2018) to compare our algorithms with the ones in (Li & Guo, 2018).

For hardware, we use a machine with 72 Intel Xeon Gold 6230 CPUs and 1TB memory. The notation "machine" used in our algorithms refers to an individual processor. In (Li & Guo, 2018), the partition of a dataset is stored in a list, and each part of the dataset is processed sequentially by a single processor to simulate the distributed environment. In our experiments, for fair comparison, we follow the settings in (Li & Guo, 2018). Although our hardware has 72 processors, we also use a single processor to handle each part of the dataset sequentially.

Algorithms and parameters. We compare our algorithm described in Algorithm 5 with other distributed algorithms. Algorithm **glz** is proposed by (Malkomes et al., 2015), which serves as the baseline for clustering cost. Algorithm **dist_kzc_0.99** is proposed by (Li & Guo, 2018) with $\epsilon = 0.99$. Algorithm **dist_kzc_0.1** is the one in (Li & Guo, 2018) with $\epsilon = 0.1$. Algorithm **ours_0.1** is our Algorithm 5 with $\epsilon = 0.1$, and algorithm **ours_0.99** is our Algorithm 5 with $\epsilon = 0.99$. In our experiments, we fix the parameter $\eta = 0.5$ and multiply the sampling rounds by a factor $\beta = 0.01$.

Experiment setup. For each parameter setting, the experiments are repeated for five times, and we take the average results. In the source code of **dist_kzc**[3] (Li & Guo, 2018), a process to guess the optimal radius of the given instance was presented using $O(\log\log\Delta)$ rounds of communication, which avoids sending $O(\frac{\log\Delta}{\epsilon})$ sets of results to the coordinator. In our experiment, we follow the operations of **dist_kzc** to avoid sending $O(\frac{\log\Delta}{\epsilon})$ sets of results to coordinator. Following the settings in (Li & Guo, 2018), the communication cost is defined as the number of representations sent by the machines multiplied by its dimension, and the final results are computed by removing only z outliers. For the (k, z)-center problem, since the radius of a solution is actually equivalent to the clustering cost, we compare the clustering radius between different algorithms to show the clustering quality of them. In (Li & Guo, 2018), different values of m were used for different datasets, where the number of machines gradually increases as the data sizes grow, which is very common in real-world applications. In our experiments, for fair comparison, we follow the same settings and use the same values of m as in (Li & Guo, 2018). For the number z of outliers, in the experiments of (Li & Guo, 2018), z is fixed to be 1024 when testing the performance of different distributed algorithms with varying number k of clusters . For fair comparison, we follow the same settings in (Li & Guo, 2018) and fix $z = 1024$ when k varies.

Results. Assume that z and m are fixed on the datasets. Figrue 1 shows the comparison results of the clustering cost, communication cost and running time with varying number of clusters k on large datasets. The comparison results of

[1] https://archive.ics.uci.edu/ml/index.php

[2] http://corpus-texmex.irisa.fr

[3] https://github.com/xyguo/clusterz

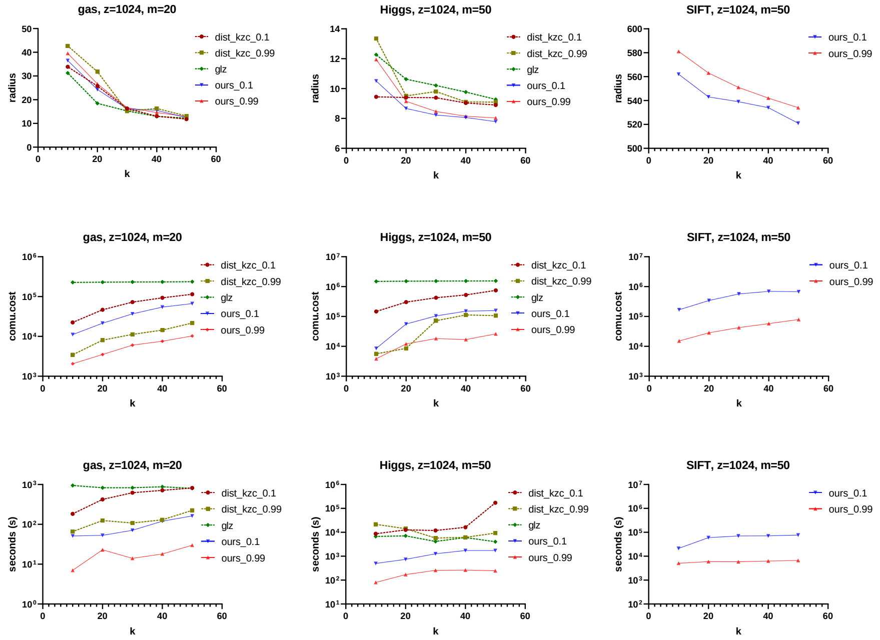

Figure 1. Comparison results of clustering performance on large datasets for (k, z)-center with varying z

the clustering cost, communication cost and running time with varying number of clusters k on small datasets are given in Figure 2 (see Appendix D). For clustering cost, our algorithms are very close to the baseline algorithm glz. The communication cost of our algorithm with $\epsilon = 0.99$ is much smaller than other algorithms. On datasets gas, covertype and Higgs, the communication cost is reduced by at least 30% compared to dist_kzc_0.99. The experiment results show that ours_0.99 runs faster than other algorithms. For dataset Higgs, ours_0.99 is more than 26 times faster than other algorithms. For dataset SIFT, it requires at least 48 hours for other algorithms to return the clustering results, and our sampling-based algorithm with $\epsilon = 0.99$ gives clustering results within 2 hours. Compared to dist_kzc, by calculating the average values of five datasets, we can get that the communication cost is reduced by 48%, and our algorithm is more than 24 times faster than other algorithms.

Assume that k and m are fixed on the datasets. The comparison results of the clustering cost, communication cost and running time with varying number of z on large datasets are given in Figure 3 (see Appendix D). The comparison results of the clustering cost, communication cost and running time with varying number of z on small datasets are given in Figure 4 (see Appendix D). Our algorithm with $\epsilon = 0.99$ gets the smallest communication cost on datasets skin, covertype and gas, where the communication cost is reduced by at least 35% compared to dist_kzc_0.99. Moreover, ours_0.99 is the fastest one compared to other algorithms. For dataset Higgs, ours_0.99 is more than 30 times faster than other algorithms. For dataset SIFT, it needs at least 48 hours for other algorithms to return the clustering results even when the number of outliers z is small, and our sampling-based algorithm with $\epsilon = 0.99$ gives clustering results within 2 hours. Compared to dist_kzc, by calculating the average values of five datasets, we can get that the communication cost is reduced by 50%, and our algorithm is more than 19 times faster than other algorithms.

7. Conclusion

In this paper, we give improved approximation algorithms for the distributed (k, z)-center problem based on sampling methods. We show that our proposed sampling-based methods can be extended to the distributed (k, z)-median/means problems. Experiments show that the proposed algorithm based on space-narrowing sampling outperforms the state-of-the-art distributed algorithms.

Acknowledgments

This work was supported by National Natural Science Foundation of China (62172446, U1909208, 61972423), Open Project of Xiangjiang Laboratory (22XJ02002, 22XJ03005), and Central South University Research Programme of Advanced Interdisciplinary Studies (2023QYJC023). This work was also carried out in part using computing resources at the High Performance Computing Center of Central South University.

References

Awasthi, P., Bakshi, A., Balcan, M., White, C., and Woodruff, D. P. Robust communication-optimal distributed clustering algorithms. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming*, pp. 18:1–18:16, 2019.

Balcan, M.-F. F., Ehrlich, S., and Liang, Y. Distributed k-means and k-median clustering on general topologies. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pp. 1995–2003, 2013.

Bhaskara, A., Vadgama, S., and Xu, H. Greedy sampling for approximate clustering in the presence of outliers. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 11146–11155, 2019.

Chakrabarty, D., Goyal, P., and Krishnaswamy, R. The non-uniform k-center problem. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming*, pp. 67:1–67:15, 2016.

Charikar, M., Khuller, S., Mount, D. M., and Narasimhan, G. Algorithms for facility location problems with outliers. In *Proceedings of the 12th Annual Symposium on Discrete Algorithms*, pp. 642–651, 2001.

Chawla, S. and Gionis, A. K-means-: A unified approach to clustering and outlier detection. In *Proceedings of the 13th SIAM International Conference on Data Mining*, pp. 189–197, 2013.

Chen, J., Azer, E. S., and Zhang, Q. A practical algorithm for distributed clustering and outlier detection. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 2253–2262, 2018.

Chen, K. A constant factor approximation algorithm for k-median clustering with outliers. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 826–835, 2008.

Cohen-Addad, V., Mirrokni, V., and Zhong, P. Massively parallel k-means clustering for perturbation resilient instances. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 4180–4201. PMLR, 2022.

Dandolo, E., Pietracaprina, A., and Pucci, G. Distributed k-means with outliers in general metrics. *arXiv preprint arXiv:2202.08173*, 2022.

Deshpande, A., Kacham, P., and Pratap, R. Robust k-means++. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence*, pp. 799–808. PMLR, 2020.

Ding, H., Liu, Y., Huang, L., and Li, J. K-means clustering with distributed dimensions. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1339–1348, 2016.

Ding, H., Yu, H., and Wang, Z. Greedy strategy works for k-center clustering with outliers and coreset construction. In *Proceedings of the 27th Annual European Symposium on Algorithms*, pp. 40:1–40:16, 2019.

Grunau, C. and Rozhoň, V. Adapting k-means algorithms for outliers. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 7845–7886, 2022.

Guha, S., Li, Y., and Zhang, Q. Distributed partial clustering. In *Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures*, pp. 143–152, 2017.

Gupta, S., Kumar, R., Lu, K., Moseley, B., and Vassilvitskii, S. Local search methods for k-means with outliers. *Proceedings of the VLDB Endowment*, 10:757–768, 2017.

Im, S., Qaem, M. M., Moseley, B., Sun, X., and Zhou, R. Fast noise removal for k-means clustering. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, pp. 456–466. PMLR, 2020.

Krishnaswamy, R., Li, S., and Sandeep, S. Constant approximation for k-median and k-means with outliers via iterative rounding. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 646–659, 2018.

Li, S. and Guo, X. Distributed k-clustering for data with heavy noise. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7849–7857, 2018.

Lloyd, S. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

Malkomes, G., Kusner, M. J., Chen, W., Weinberger, K. Q., and Moseley, B. Fast distributed k-center clustering with outliers on massive data. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pp. 1063–1071, 2015.

Zhang, Z., Feng, Q., Huang, J., Guo, Y., Xu, J., and Wang, J. A local search algorithm for k-means with outliers. *Neurocomputing*, 450:230–241, 2021.

A. Missing Proofs in Section 3

Lemma 3.1. *By executing Algorithm 1 on machine i, with probability at least $1-\eta$, the set S_i sampled in step 3 of Round 1 in Algorithm 1 contains at least one point from $P_{opt} \cap P_i$.*

Proof. Define $\zeta = \frac{|P_i \cap P_{opt}|}{|P_i|}$. Since $|P_i| \geq (1+\epsilon)z$, it holds that $\zeta \geq \frac{\epsilon}{1+\epsilon}$. By randomly sampling a set S_i from P_i, the probability that S_i contains at least one data point from $P_{opt} \cap P_i$ is at least $1-(1-\zeta)^{|S_i|}$. In order to guarantee that, with probability at least $1-\eta$, at least one data point is sampled from $P_i \cap P_{opt}$, we have $1-(1-\zeta)^{|S_i|} \geq 1-\eta$. Thus, S_i has size at least $\frac{\log \frac{1}{\eta}}{\log \frac{1}{1-\zeta}} \leq \frac{1}{\zeta} \log \frac{1}{\eta}$. Since $\zeta \geq \frac{\epsilon}{1+\epsilon}$, if $|S_i| \geq \frac{1+\epsilon}{\epsilon} \log \frac{1}{\eta}$, then S_i contains at least one point from $P_i \cap P_{opt}$ with probability at least $1-\eta$. □

Lemma 3.2. *In each iteration j of the for loop in step 4 of Round 1 in Algorithm 1 on machine i, either $d(R_{i,j}, Q_{i,j-1}) \leq 2L^*$ or $|\beta_i(Q_{i,j})| > |\beta_i(Q_{i,j-1})|$ with probability at least $1-\eta$.*

Proof. In the j-th iteration of the for loop in step 4 of Round 1 in inliers-recalling sampling, let $R_{i,j}$ be the set of the furthest $(1+\epsilon)z$ points from $Q_{i,j-1}$ in step 5. We have the following two cases: (1) $d(R_{i,j}, Q_{i,j-1}) \leq 2L^*$; (2) $d(R_{i,j}, Q_{i,j-1}) > 2L^*$. For case (2), let F be the set of optimal clusters that are not covered by the points in $Q_{i,j-1}$ with radius $2L^*$, and let $P(F)$ be the set of data points contained in clusters of F with distances larger than $2L^*$ to $Q_{i,j-1}$. Since $d(R_{i,j}, Q_{i,j-1}) > 2L^*$, we have $(R_{i,j} \cap P_{opt}) \subseteq P(F)$. According to Lemma 3.1, by randomly sampling $\frac{1+\epsilon}{\epsilon} \log \frac{1}{\eta}$ points from $R_{i,j}$, the sampled set contains at least one point from $P(F)$ with probability at least $1-\eta$, which makes at least one uncovered optimal cluster covered by data points in $Q_{i,j}$ with radius $2L^*$. Thus, we can get that in the j-the iteration of the for loop in step 4 of Round 1 in Algorithm 1, either $d(R_{i,j}, Q_{i,j-1}) \leq 2L^*$ or $|\beta_i(Q_{i,j})| > |\beta_i(Q_{i,j-1})|$ with probability at least $1-\eta$. □

Lemma 3.4. *For each machine $i \in [m]$, assume that $d(R_{i,j}, Q_{i,j-1}) > 2L^*$ holds for each $j \in [\lambda]$ during the sampling process in steps 4-7 of Round 1 in Algorithm 1, where $\lambda = \gamma k \log m$, and γ is a large constant. Then, with probability at least $1-\frac{\eta}{m}$, $\beta_i(Q_{i,\lambda}) = k$.*

Proof. Consider a single machine $i \in [m]$. In inliers-recalling sampling, let $Q_{i,j}$ be the set of centers found after j iterations of the for loop in step 4 of Round 1 in Algorithm 1, and let $\beta_i(Q_{i,j}) \subseteq [k]$ denote the set of indices of optimal clusters covered by the points in $Q_{i,j}$ with radius $2L^*$. We define a variable $a_j = 1$ if $|\beta_i(Q_{i,j})| > |\beta_i(Q_{i,j-1})|$. Otherwise $a_j = 0$. Since we assume that $d(R_{i,j}, Q_{i,j-1}) > 2L^*$ holds on each machine, a_j is equal to 1 with probability at least $1-\eta$ for each j by Lemma 3.2. Let t be the number of iterations of the for loop in step 4 of Round 1 in Algorithm 1 such that $t \geq \frac{2\delta k \ln \frac{m}{\eta}}{1-\eta}$, where δ is a constant with $\delta \geq 4$. Then, by Chernoff Bounds (Lemma 3.3), we have $P_r\left(\sum_{j=1}^{t} a_j < k\right) < P_r\left(\sum_{j=1}^{t} a_j < \frac{1}{2}(1-\eta)t\right) < e^{-\frac{t(1-\eta)}{8}} \leq e^{-\frac{\delta \ln \frac{m}{\eta}}{4}}$, where the first inequality follows from the assumption that $k \leq \frac{1}{2}(1-\eta)t$, the second inequality follows from Lemma 3.2 and the fact that $E\left(\sum_{j=1}^{t} a_j\right) \geq (1-\eta)t$, and the last inequality follows from the assumption that $t \geq \frac{2\delta}{1-\eta} \ln \frac{m}{\eta}$. This implies that $P_r(\beta_i(Q_i) = k) = 1 - P_r\left(\sum_{j=1}^{t} a_j < k\right) > 1 - e^{-\frac{\delta \ln \frac{m}{\eta}}{4}} \geq 1 - \frac{\eta}{m}$, where the last inequality follows from the assumption that $\delta \geq 4$. Thus, given that η and δ are constants, by repeating the sampling process for $O(k \log m)$ times, with probability at least $1-\frac{\eta}{m}$, we have $\beta_i(Q_i) = k$. □

Lemma 3.7. *In the j-th iteration of Algorithm 2, let c_j be the center added to C' in step 6. Then, for each uncovered optimal cluster D_h^*, $\sum_{p \in B_{Y_j}(c_j, 6L)} w(p) \geq |U_{D_h^*}|$.*

Proof. Let H be the set of points discarded across machines by P_{L_f} with radius L_f, and let $X = P \backslash H$. Consider an optimal cluster D_h^* $(1 \leq h \leq k)$. Let p, q be two arbitrary points in $D_h^* \cap X$. We first bound the distance between s_p and s_q. By Corollary 3.5, we have $d(p, s_p) \leq L_f \leq 2L^*$ and $d(q, s_q) \leq L_f \leq 2L^*$. According to the triangle inequality, $d(s_p, s_q) \leq d(s_p, p) + d(p, q) + d(q, s_q) \leq 2L_f + d(p, d_h^*) + d(d_h^*, q) \leq 6L$, where $L \in [L^*, (1+\epsilon)L^*]$. For any point $u \in D_h^*$ with $s_u \in Y_j$, $B_{P_{L_f}}(s_u, 6L)$ contains all the representation points in $D_h^* \cap X$. Note that our algorithm always chooses a point $c_j \in Y_j$ with maximum summation of the weights in $B_{Y_j}(c_j, 6L)$. Hence, for each uncovered optimal cluster D_h^*, $\sum_{p \in B_{Y_j}(c_j, 6L)} w(p) \geq |U_{D_h^*}|$. □

Lemma 3.8. *In the j-th iteration of Algorithm 2, let $\lambda_j = B_{Y_j}(c_j, 12L)$. Then, $\sum_{j=1}^{k} \sum_{p \in \lambda_j} w(p) \geq |P_{opt} \cap X|$.*

Proof. We prove the lemma by induction based on the following claim: After j iterations of the step 7 in Algorithm 2, there exists an ordering $D_{t_1}^*, \ldots, D_{t_j}^*$ of optimal clusters such that the summation of the weights in P_{L_f} covered by the points in C' is at least $|\bigcup_{h=1}^{j}(D_{t_h}^* \cap X)|$. By Lemma 5, for $j = 1$, the claim is correct. Assume that before iteration j, there exists an ordering $D_{t_1}^*, \ldots, D_{t_{(j-1)}}^*$ such that the summation of the weights in P_{L_f} covered by the points in C_{L_f} is at least $|\bigcup_{h=1}^{j-1}(D_{t_h}^* \cap X)|$, and each point in $\bigcup_{h=1}^{j-1}(D_{t_h}^* \cap X)$ has been mapped to a unique weight unit of points in $\cup_{h=1}^{j-1}\lambda_j$. Let $G_j = B_{Y_j}(c_j, 6L)$. For each iteration of Algorithm 2, we have the following two cases: (1) there exists an optimal cluster D_f^* ($f \in [k] \backslash \{t_1, \ldots, t_{(j-1)}\}$) such that at least one point $p \in D_f^*$ can be found with its representation s_p intersecting with $\cup_{i=1}^{j} G_j$, i.e., $s_p \in \cup_{i=1}^{j} G_j$; (2) for any index ($f \in [k] \backslash \{t_1, \ldots, t_{(j-1)}\}$), no representation of the points in D_f^* is contained in $\cup_{i=1}^{j} G_j$.

For case (1), by Lemma 5, for any point $q \in D_h^* \cap X$, $d(s_q, c_j) \leq d(s_q, s_p) + d(s_p, c_j) \leq 12L$. Hence, the representations of the points in $D_h^* \cap X$ are all contained in $B_{Y_j}(c_j, 12L)$. Let $t_j = f$. We now map each data point in $p \in D_f^*$ to a weight unit of its representation s_p, and the unit weight can be viewed as the weight that p contributes to s_p.

For case (2), let D_f^* be an arbitrary optimal cluster such that $f \in [k] \backslash \{t_1, \ldots, t_{(j-1)}\}$. Let $V_{D_f^*} = \left\{p \in D_f^* \cap X : s_p \in \cup_{i=1}^{j-1} \lambda_j\right\}$ denote the set of points in D_f^* whose representations are already covered before the j-th iteration. Denote $U_{D_f^*} = (D_f^* \cap X) \backslash V_{D_f^*}$ as the set of points whose representations are uncovered. Similar to case (1), we map each point in $V_{D_f^*}$ to a weight unit of its representation. For the data points in $U_{D_f^*}$, by Lemma 5, we get that $\sum_{p \in G_j} w_p \geq |U_{D_f^*}|$, and there exists a mapping such that no two points in $U_{D_f^*}$ are mapped to the same weight unit.

For both cases, each point must be mapped to a unique unit of weight. For the first case, all the representations of data points in D_f^* are not in Y_j after the j-th iteration, and points in D_f^* will not be used for mapping again. Since each point is mapped to a weight unit of its representation, no two points in D_f^* are mapped to the same weight unit. For the second case, similar to case (1), the representations of data points in $V_{D_f^*}$ are not in Y_j after the j-th iteration, and points in $V_{D_f^*}$ will not be used for mapping again. For data points in $U_{D_f^*}$, they are mapped to the weight units of the points in G_j. After the j-th iteration, G_j is removed from Y_j. Hence, no data point in D_f^* where $f \in [k] \backslash \{t_1, \ldots, t_{(j-1)}\}$ is mapped to G_j in case (2). Note that G_j does not intersect with any optimal cluster D_f^*, where $f \in [k] \backslash \{t_1, \ldots, t_{(j-1)}\}$, and the points in G_j will not be mapped again in case (1). Based on $D_{t_1}^*, \ldots, D_{t_{(j-1)}}^*$ and D_f^*, we can find an ordering $D_{t_1}^*, \ldots, D_{t_j}^*$ of optimal clusters with $t_j = f$ such that the summation of the weights in P_{L_f} covered by the points in C' is at least $|\bigcup_{h=1}^{j}(D_{t_h}^* \cap X)| = |P_{opt} \cap X|$. □

Theorem 1.1. *For the distributed (k, z)-center problem, there exists a 4-round distributed algorithm that outputs a $(14(1+\epsilon), 1+\epsilon)$-approximate solution with constant probability, where the communication cost is $O(\frac{m^3 k \log(mk)}{\epsilon^2})$, and the running time on each machine i is $O(\frac{n_i m^3 k \log(mk)}{\epsilon^2})$.*

Proof. By Corollary 3.6 and Corollary 3.9, the communication cost and the number of outliers discarded are $O(\frac{m^3 k \log(mk)}{\epsilon^2})$ and $(1+\epsilon)z$, respectively. For any given instance of the distributed (k, z)-center problem, let C' be the set of centers obtained with minimum cost. For each point p in P whose representation point is covered by C', by triangle inequality, we have $d(p, C') \leq d(p, s_p) + d(s_p, C') \leq L_f + 12L \leq 14L$, where $L \in [L^*, (1+\epsilon)L^*]$. Then a $(14(1+\epsilon), (1+\epsilon))$-approximate solution can be obtained with probability at least $(1-\eta)^2$. For inliers-recalling sampling, the sampling process takes $O(1)$ time, and the furthest $(1+\epsilon)z$ and the nearest $\frac{\epsilon z}{2m}$ data points can be found in $O(n_i)$ time using linear selection algorithm in (Ding et al., 2019). Thus, the running time on each machine for Round 1 of Algorithm 1 and Round 3 of Algorithm 1 are $O(\frac{n_i m^2 k \log(mk)}{\epsilon^2})$ and $O(\frac{n_i m^3 k \log(mk)}{\epsilon^2})$, respectively. Hence, the total running time on each machine is $O(\frac{n_i m^3 k \log(mk)}{\epsilon^2})$. □

B. Missing Proofs in Section 4

Lemma 4.1 Let $F \subseteq [m]$ denote the set of machines where case (2) happens at least once during the space-narrowing sampling process. Then, the total number of uncovered inliers is bounded by ϵz.

Proof. For each machine $i \in F$, in Algorithm 3, let $r_{i,j}$ be the integer found before step 7 of Algorithm 3 when case (2) happens at the first time on machine i. Since case (2) happens on machine i, i.e., $|Z^* \cap U_{i,j-1}| \geq \frac{\epsilon_1 r_{ij} z}{m}$, $|P_{opt} \cap U_{i,j-1}|$ is at most $\frac{(1+\epsilon_1)\epsilon_1(r_{ij}+1)z}{m} - \frac{\epsilon_1 r_{ij} z}{m}$. Therefore, the total number of uncovered inliers among m machines is bounded by $\sum_{i\in F}\left(\frac{(1+\epsilon_1)\epsilon_1(r_{ij}+1)z}{m} - \frac{\epsilon_1 r_{ij} z}{m}\right) = \sum_{i\in F}\left(\frac{\epsilon_1^2 r_{ij} z}{m} + \frac{(1+\epsilon_1)\epsilon_1 z}{m}\right)$. Observe that there are only z outliers and $|Z^* \cap U_{i,j-1}| \geq \frac{\epsilon_1 r_{ij} z}{m}$, which means $\sum_{i\in F}\frac{\epsilon_1 r_{ij} z}{m} \leq \sum_{i\in F}|Z^* \cap U_i| \leq z$. Then, the total number of uncovered inliers is bounded by $\epsilon_1 z + (1+\epsilon_1)\epsilon_1 z$. Together with the fact that $\epsilon \in (0, 1]$ and $\epsilon_1 = \frac{\epsilon}{3}$, the total number of uncovered points is bounded by ϵz. □

Theorem 1.2. *For the distributed (k, z)-center problem, there exists a 2-round distributed algorithm that outputs a $(14(1+\epsilon), 1+\epsilon)$-approximate solution with constant probability, where the communication cost is $O(\frac{mk\log m}{\epsilon} \cdot \frac{\log\Delta}{\epsilon})$, and the local running time on each machine i is $O(\frac{n_i k\log m}{\epsilon} \cdot \frac{\log\Delta}{\epsilon})$.*

Proof. By Corollary 3.9 and Corollary 4.2, the communication cost and the number of outliers discarded are $O(\frac{mk\log m}{\epsilon} \cdot \frac{\log\Delta}{\epsilon})$ and $(1+\epsilon)z$, respectively. For any given instance of the distributed (k, z)-center problem, by calling Algorithm 5 for each possible values of L, let C' be the set of centers obtained with minimum cost. For each point p in P whose representation point is covered by C', by triangle inequality, we have $d(p, C') \leq d(p, s_p) + d(s_p, C') \leq 12L + 2L = 14L$. If $L \in [L^*, (1+\epsilon)L^*]$, then a $(14(1+\epsilon), (1+\epsilon))$-approximation solution can be obtained with probability at least $1 - \eta$. In space-narrowing sampling, the sampling process takes $O(1)$ time, and the ball coverage for the removal of data points takes time $O\left(\frac{(1+\epsilon)n_i}{\epsilon}\right)$. By repeating the whole process $O(k\log m)$ times, the total running time on each machine is $O(\frac{n_i k\log m\log\Delta}{\epsilon^2})$. □

C. Extension to the k-Median/Means Problems with Outliers

In this section, we consider the (k, z)-median/means problems in distributed setting. The goal of the (k, z)-median/means problems is to find a set $C \subseteq P$ of k centers and a set $Z \subseteq P$ of z outliers such that the corresponding clustering cost on $P\backslash Z$ with respect to C is minimized. For the (k, z)-median problem, the clustering cost is defined as the sum of the distances from data points in $P\backslash Z$ to their closest centers in C, i.e., $\sum_{p\in P\backslash Z} d(p, C)$. For the (k, z)-means problem, the clustering cost is defined as the sum of the squared distances from data points in $P\backslash Z$ to their closest centers in C, i.e., $\sum_{p\in P\backslash Z} d^2(p, C)$. For two sets A, $B \subseteq P$, we use $\Psi(A, B) = \sum_{x\in A} d(x, B)$ to denote the summation of the distances from data points in A to their closest data points in B. In the following, we take the distributed (k, z)-median problem as an example for analysis. By using a relaxed triangle inequality (Chen et al., 2018), the results for distributed (k, z)-median can be easily extended to distributed (k, z)-means. Given an instance of distributed (k, z)-median, for an optimal cluster $D_h^* \in D^*$, we use $OPT_h = \Psi(P_h^*, C^*)$ to denote the optimal clustering cost of D_h^*. Define $OPT = \sum_{i=1}^{k} OPT_i$ as the optimal clustering cost. Let $C_f \subseteq P$ be a set of centers. We use $\Psi^{-(1+\epsilon)z}(P, C_f)$ to denote the clustering cost of data points in $P\backslash Z_f$ to C_f, where Z_f is the set of the furthest $(1+\epsilon)z$ data points in P to C_f.

C.1. Distributed (k, z)-Median/Means Algorithm by Inliers-Recalling Sampling Method

In this subsection, we show how to apply our proposed inliers-recalling sampling method to obtain an $O(\frac{1}{\epsilon^2})$-approximate solution with $(1+\epsilon)z$ outliers discarded and near-linear local running time in the data size, where the communication cost is independent of the aspect ratio. The general idea behind is to firstly use a filtering process to obtain a bi-criteria solution with $(O(1), O(1))$-approximation on each machine such that the number of data points discarded can be bounded by $O(\frac{z}{m})$ if the data points are randomly partitioned among machines. Denote the set of outliers discarded on machine i as Z_i. By the filtering process, it can be guaranteed that the inliers take a large fraction of the data points in Z_i. Then, by applying our proposed inliers-recalling sampling algorithm (Algorithm 1) on Z_i, most inliers can be recalled back, which ensures that there are at most ϵz uncovered inliers across the machines.

Algorithm 6 Distributed (k, z)-median/means by IRS

```
Input: A partition $\{P_1, P_2, ..., P_m\}$ of dataset $P$ among $m$ machines, and parameters $k$, $z$, $\eta$, $\epsilon$.
Output: A set $C \subseteq P$ of size at most $k$.
Rounds 1-2
Call the algorithm in (Chen et al., 2018) by setting $z = \frac{(1+\epsilon)z}{m}$ to obtain an $O(1)$-approximate solution $S_i$ on $P_i$ with $\frac{8(1+\epsilon)z}{m}$ outliers discarded, and denote the set of outliers discarded as $Z_i$.
Execute Rounds 1-2 of IRS($\{Z_1, Z_2, ..., Z_m\}$, $k$, $(1 + \frac{\epsilon}{2})z$, $\eta$, $\frac{\epsilon}{2}$).
Let $Q_i$ be the candidate set of centers obtained by executing Rounds 1-2 of IRS on $Z_i$.

Round 3 on each machine $i \in [m]$
Let $L'$ be the collection of radii sent from the coordinator.
for $L \in L'$ do
    $H_s = \cup_{q \in Q_i} B_{Z_i}(q, L)$, $Q' = Q_i$, $S' = S_i$.
    Assign each data point in $H_s$ to its closest center in $Q'$.
    For each $q \in Q'$, let $\sigma(q)$ be the number of points assigned to $q$, and assign a weight of $\sigma(q)$ to $q$.
    Assign each data point in $P_i \backslash Z_i$ to its closest center in $S'$.
    For each $s \in S'$, let $\sigma(s)$ be the number of points assigned to $s$, and assign a weight of $\sigma(s)$ to $s$.
    $S_i^L = Q' \cup S'$.
    Send $(S_i^L, L)$ to the coordinator.
end for

Round 4 on the coordinator.
Initialize $C_f = \emptyset$, $L_f = \infty$.
for $L \in L'$ do
    $P_L = \cup_{i=1}^m S_i^L$.
    For each point $p$ in $P_L$, let $w(p)$ be the weight of $p$.
    $z' = (1 + \epsilon)z + \sum_{p \in P_L} w(p) - |P|$.
    if $z' > 0$ then
        Call a weighted clustering with outliers algorithm on $P_L$ with $z = z'$, and denote the set of centers returned as $C_t$.
        if $\Psi^{-(1+\epsilon)^2 z}(P, C_t) < L_f$ then
            $L_f = \Psi^{-(1+\epsilon)^2 z}(P, C_t)$, $C_f = C_t$.
        end if
    end if
end for
Return $C_f$.
```

Following the settings in (Chen et al., 2018), we assume that $\frac{z}{m} \geq \Omega(\log n)$, i.e., $\frac{z}{m} \geq \frac{3}{\epsilon^2} \log \frac{n}{\eta}$, where parameters ϵ and $\eta \in (0, 1]$ are used to control the number of outliers discarded and the success probability, respectively. As pointed out in (Chen et al., 2018), this assumption is justifiable in practice since the number of outliers z typically scales with the size of the dataset while the number of machines m is usually a fixed number. Consider a single machine $i \in [m]$, for each outlier $z \in Z^*$, define $X_z^i = 1$ if z is assigned to machine i. Otherwise define $X_z^i = 0$. The outliers in Z^* are called true outliers. According to the random partition rule, it holds that $P_r(X_z^i = 1) = \frac{1}{m}$ for each $z \in Z^*$, which means $E[\sum_{z \in Z^*} X_z^i] = \frac{z}{m}$, where $E[X]$ denote the expectation of the random variable X. By applying the Chernoff Bound (Lemma 3.3), we can get that with probability at least $1 - e^{-\frac{\frac{z}{m}\epsilon^2}{3}} \geq 1 - \frac{\eta}{n}$, the number of true outliers assigned on machine i can be bounded by $(1 + \epsilon)\frac{z}{m}$. By taking a union bound over all machines, we have that the number of true outliers assigned on each machine $i \in [m]$ can be bounded by $(1 + \epsilon)\frac{z}{m}$ with probability at least $1 - \eta$. In the following, we assume that the number of true outliers assigned on each machine is upper bounded by $\frac{(1+\epsilon)z}{m}$.

Theorem 5.3 (Chen et al., 2018) *For the (k, z)-median/means problems, there exists an approximation algorithm that outputs a $(O(1), 8)$-approximate solution with high probability by opening $O(k \log n)$ centers, where the running time is $O(\max\{k, \log n\}n)$.*

By Theorem 5.3, we can obtain an $O(1)$-approximate solution with $\frac{8(1+\epsilon)z}{m}$ outliers discarded on each machine $i \in [m]$.

The formal algorithm solving the distributed (k, z)-median/means by inliers-recalling sampling is given in Algorithm 6. In the following, we focus on the distributed (k, z)-median problem to analyze Algorithm 6. Let $P' = \{p \in P_{opt} : d(p, C^*) > \frac{2OPT}{\epsilon z}\}$ be the set of the inliers with distances larger than $\frac{2OPT}{\epsilon z}$ to their optimal clustering centers. Observe that $|P'| \leq \frac{\epsilon z}{2}$. Otherwise, the clustering cost of the data points in P' to C^* is at least OPT, which contradicts with the fact that $\Delta(P', C^*) \leq OPT$. We can view the data points in P' as the set of additional outliers to be discarded since they are far from optimal clustering centers. By replacing z with $(1 + \frac{\epsilon}{2})z$ and ϵ with $\frac{\epsilon}{2}$ in the input of inliers-recalling sampling algorithm (Algorithm 1) and executing the inliers-recalling sampling on Z_i, we show that there are at most ϵz inliers discarded across machines.

Now, we consider executing step 2 in rounds 1-2 of Algorithm 6, in which Algorithm 1 is called to find a collection of radii. On each machine $i \in [m]$, in step 2 of Round 1 in Algorithm 1, we have $|U_i| \geq (1 + \frac{\epsilon}{2})(1 + \frac{\epsilon}{2})z$, where data points in $P_{opt}\backslash P'$ take a fraction of at least $\frac{(1+\frac{\epsilon}{2})(1+\frac{\epsilon}{2})z - z - \frac{\epsilon}{2}z}{(1+\frac{\epsilon}{2})(1+\frac{\epsilon}{2})z} = \frac{\epsilon}{2+\epsilon}$. Hence, in step 3 of Algorithm 1, by randomly taking a subset $S_i \subseteq P_i$ with size $\frac{1+\frac{\epsilon}{2}}{\frac{\epsilon}{2}} \log \frac{1}{\eta} = \frac{2+\epsilon}{\epsilon} \log \frac{1}{\eta}$ from P_i, we can sample at least one data point $q \in P_{opt}\backslash P'$ with probability at least $1 - \eta$. Assume that $q \in D_j^*$ for some $D_j^* \in D^*$. Then, data points in $D_j^*\backslash P'$ can be covered by q with radius $\frac{4OPT}{\epsilon z}$. Similarly, in each step 6 of Algorithm 1, with probability at least $1 - \eta$, an optimal cluster D_j^* can be covered by the data points sampled with radius $\frac{4OPT}{\epsilon z}$ using Lemma 3.1. Then, by Lemma 3.4, at most $(1 + \frac{\epsilon}{2})(1 + \frac{\epsilon}{2})z$ data points are discarded as outliers on each machine $i \in [m]$.

Next, we consider the sampling process with two stages in steps 10-15 of Round 1 in Algorithm 1. Let R_i be the set of the furthest $(1 + \frac{\epsilon}{2})(1 + \frac{\epsilon}{2})z$ data points from Q_i in step 9 of Algorithm 1. The goal of the first stage (step 10 of Round 1 in Algorithm 1) is to make the large optimal clusters with $|(D_j^*\backslash P') \cap R_i| \geq \frac{\epsilon(1+\frac{\epsilon}{2})z}{4km}$ covered by sampling at least one data point from $D_j^*\backslash P'$. For each large optimal cluster D_j^*, we have $\frac{|D_j^*\backslash P'|}{|R_i|} \geq \frac{\frac{\epsilon(1+\frac{\epsilon}{2})z}{4km}}{(1+\frac{\epsilon}{2})(1+\frac{\epsilon}{2})z} = \frac{\epsilon}{4(1+\frac{\epsilon}{2})km}$. This indicates that by randomly taking a set $V_i \subseteq R_i$ of data points such that $|V_i| = \frac{4km(1+\frac{\epsilon}{2})}{\epsilon} \log \frac{km}{\eta}$, with probability at least $1 - \frac{\eta}{m}$, we can make each large optimal cluster D_j^* covered by data points in V_i with radius $\frac{4OPT}{\epsilon z}$. Then, in the second stage (steps 11-15 of Round 1 in Algorithm 1), a recalling process is used to iteratively find the discarded inliers back. The analysis is similar to the case for the distributed (k, z)-center problem. By Corollary 3.5, with probability at least $1 - \eta$, the increase on clustering cost by recalling data points can be bounded by $O(\frac{OPT}{\epsilon m})$ on each machine, where at most $\frac{\epsilon(1+\frac{\epsilon}{2})z}{4m}$ inliers are discarded. In the second round of inliers-recalling sampling algorithm, each machine sends a list of radii L_i to the coordinator, and we can obtain a collection L' of radii with size $|L'| = O(\frac{m^2}{\epsilon})$. Let Q_i be the set of centers obtained in step 10 of Round 1 in Algorithm 1. By Corollary 3.5, it can be seen that there exists at least one radius $L_f \in L'$ such that at most $\frac{\epsilon}{2}(1 + \frac{\epsilon}{2})z$ inliers are discarded across the machines using each Q_i as the set of centers with radius L_f. Note that there are at most $8(1 + \epsilon)\frac{z}{m}$ data points in Z_i on each machine. Hence, the increase on clustering cost can be bounded by $8(1 + \epsilon)\frac{z}{m} \cdot \frac{4OPT}{\epsilon z} = O(\frac{OPT}{\epsilon m})$ on each machine even if all the data points are covered by balls with radius $L = \frac{4OPT}{\epsilon z}$. By taking a summation over all machines, the increase on clustering cost can be bounded by $O(\frac{OPT}{\epsilon})$.

As discussed above, by applying the inliers-recalling sampling method, we can obtain a set P_{L_f} of weighted representations in step 3 of Round 4 in Algorithm 6 with approximation ratio $O(\frac{1}{\epsilon})$ and at most ϵz uncovered inliers across the machines for some $L_f \in L'$. Denote the set of outliers discarded across machines as R_f. More formally, we have $R_f \cap P_{opt} \leq \frac{\epsilon}{2}(1 + \frac{\epsilon}{2})z \leq \epsilon z$, and P_{L_f} induces an $O(\frac{1}{\epsilon})$-approximation, i.e., $\sum_{p \in P\backslash R_f} d(p, P_{L_f}) \leq O(\frac{1}{\epsilon})OPT$. By executing a weighted (k, z)-median/means approximation algorithm, such as the local search algorithm given in (Grunau & Rozhoň, 2022), that achieves an $O(\frac{1}{\epsilon})$-approximation with $(1 + \epsilon)z$ outliers discarded and replacing z with $(1 + \epsilon)z - |R_f|$, we prove that an $O(\frac{1}{\epsilon^2})$-approximate solution with $(1 + \epsilon)^2 z$ outliers discarded can be obtained. In this case, a set Z_f of data points with size $(1 + \epsilon)((1 + \epsilon)z - |R_f|)$ is discarded by the weighted algorithm. Hence, the total number of outliers discarded can be bounded by $|R_f \cup Z_f| \leq (1 + \epsilon)^2 z$. By replacing ϵ with $\frac{1}{3}\epsilon$, the total number of data points discarded can by bounded by $(1 + \epsilon)z$ using the fact that $\epsilon \leq 1$.

Lemma 5.4. *Let P_L be a collection of weighted representations with $O(\frac{1}{\epsilon})$-approximation on the coordinator in step 3 of Round 4 in Algorithm 6, where at most ϵz inliers are discarded across the machines. Let R_f be the set of data points discarded across the machines. Suppose that there is a weighted (k, z)-median/means algorithm A that achieves an $O(\frac{1}{\epsilon})$-approximation with $(1 + \epsilon)z$ outliers discarded. By executing algorithm A on P_L and setting the number of outliers discarded as $z = (1 + \epsilon)z - |R_f|$, we can obtain a $(O(\frac{1}{\epsilon^2}), 1 + \epsilon)$-approximate solution.*

Proof. Let C_f be the set of centers returned by Algorithm A. Based on C_f, let Z_f be the set of the furthest $(1+\epsilon)((1+\epsilon)z - |R_f|)$ data points in $P\backslash R_f$ to C_f. For each data point $x \in P$, we use $s_x \in P_L$ to denote the data point in P_L with the smallest distance to x. For each data point $x \in P_L$, by assigning each data point in $P\backslash(R_f \cup Z_f)$ to its closest center in P_L, we use $w(x)$ to denote the number of data points assigned to x. Let $Z_g = Z^* \cup R_f$. Note that $|Z_g| \le (1+\epsilon)z$ since there are at most ϵz inliers in R_f, which means $|Z_g| \le |R_f \cup Z_f|$. For each data point $x \in P_L$, by assigning data points in $P\backslash Z_g$ to its closest center in P_L, we use $v(x)$ to denote the number of data points assigned to x. Observe that

$$\begin{aligned}
\Psi^{-(1+\epsilon)^2 z}(P, C_f) &\le \Psi(P\backslash(Z_f \cup R_f), C_f) \le \sum_{x\in P\backslash(Z_f\cup R_f)} d(x, s_x) + \sum_{x\in P\backslash(Z_f\cup R_f)} d(s_x, C_f) \\
&\le \sum_{x\in P\backslash R_f} d(x, s_x) + \sum_{x\in P\backslash(Z_f\cup R_f)} d(s_x, C_f) = \sum_{x\in P\backslash R_f} d(x, s_x) + \sum_{x\in P_L} w(x) d(x, C_f) \\
&\le O(\frac{1}{\epsilon})OPT + O(\frac{1}{\epsilon}) \sum_{x\in P_L} v(x) d(x, C^*) \\
&\le O(\frac{1}{\epsilon})OPT + O(\frac{1}{\epsilon}) \sum_{x\in P\backslash Z_g} d(x, s_x) + d(x, C^*) \\
&\le O(\frac{1}{\epsilon})OPT + O(\frac{1}{\epsilon})(O(\frac{1}{\epsilon})OPT + OPT) = O(\frac{1}{\epsilon^2})OPT,
\end{aligned}$$

where the second inequality follows from the triangle inequality, the fourth inequality follows from the facts that the set P_L of weighted representations induces an $O(\frac{1}{\epsilon})$-approximation, $|Z_g| \le |R_f \cup Z_f|$ and Algorithm A gives an $O(\frac{1}{\epsilon})$-approximate solution, the fifth inequality follows from the triangle inequality, and the sixth inequality follows from the fact that the set P_L of weighted representations induces an $O(\frac{1}{\epsilon})$-approximation. By replacing ϵ with $\frac{\epsilon}{3}$, the number of outliers discarded can be bounded by $(1+\epsilon)z$. Together with Corollary 3.5 and Theorem 5.3, the total communication cost of Algorithm 6 is $O(\frac{m^3 k \log n \log(mk)}{\epsilon^2})$, and the running time on each machine is $O(\frac{n_i m^3 k \log(mk) \max\{\log n_i, k\}}{\epsilon^2})$. □

Putting all things together, Theorem 5.1 can be proved.

C.2. Distributed (k, z)-Median/Means Algorithm by Space-Narrowing Sampling Method

In this subsection, we propose a more practical algorithm for the distributed (k, z)-median/means problems with smaller communication rounds and communication cost independent of the number of outliers under the assumption that data points are randomly partitioned across the machines. The formal algorithm is given in Algorithm 7. As discussed in section C.1, we assume that the true outliers assigned on each machine can be bounded by $\frac{(1+\epsilon)z}{m}$. By applying the bi-criteria approximation scheme given in (Chen et al., 2018) as a filtering process and replacing z with $(1+\epsilon)\frac{z}{m}$, we can obtain an $O(1)$-approximate solution with $8(1+\epsilon)\frac{z}{m}$ outliers discarded on each machine by Theorem 5.3. However, in the worst case, the discarded $8(1+\epsilon)\frac{z}{m}$ outliers on each machine are all inliers. Hence, in step 2 of Round 1 in Algorithm 7, we can apply our space-narrowing sampling method on the discarded outliers to reduce the number of inliers discarded.

In the process of space narrowing sampling in Algorithm 7, the coverage radius is set to be $\frac{2L}{\epsilon z}$. As pointed out in (Li & Guo, 2018), a radius L can be obtained from guessing such that $OPT \le L \le (1+\epsilon)OPT$ by losing a factor of $O(\frac{\log \Delta}{\epsilon})$ on communication cost and running time. Since there are at most $8(1+\epsilon)\frac{z}{m}$ outliers discarded on each machine, the clustering cost is be increased by a factor of $8(1+\epsilon)\frac{z}{m} \cdot \frac{4L}{\epsilon z} = O(\frac{OPT}{\epsilon m})$ on each machine even if all the data points are covered by balls with radius $\frac{4L}{\epsilon z}$. By taking a summation of approximation loss over all machines, we can obtain a set of weighted representations on the coordinator with $O(\frac{1}{\epsilon})$-approximation. Next, we show that the overall number of inliers discarded across machines can be bounded by ϵz. Let $P' = \{p \in P_{opt} : d(p, C^*) > \frac{2OPT}{\epsilon z}\}$ be the set of inliers with distances larger than $\frac{2OPT}{\epsilon z}$ to their optimal clustering centers. Observe that $|P'| \le \frac{\epsilon z}{2}$. Otherwise, the clustering cost of the data points in P' to C^* is at least OPT, which contradicts with the fact $\Delta(P', C^*) < OPT$. In the following, we consider executing the space-narrowing sampling method in step 2 of Round 1 in Algorithm 7 on a single machine $i \in [m]$. For an optimal cluster $D_h^* \in D$, let $q \in D_h^*$ be an arbitrary data point in $D_h^*\backslash P'$. Note that q can cover all the data points in $D_h^*\backslash P'$ with radius $\frac{4L}{\epsilon z}$ since $d(p, q) \le d(p, c_i^*) + d(c_i^*, q) \le \frac{4OPT}{\epsilon z}$ for each $p \in D_h^*\backslash P'$. Then, in space narrowing sampling, the goal is to iteratively sample a data point q from $Z_i\backslash P'$ to cover most discarded inliers with radius $\frac{4L}{\epsilon z}$. Since during the sub-sampling process, data points in P' may not be covered, we can view them as additional outliers. Hence, by replacing z with $(1+\frac{\epsilon}{2})z$ and ϵ with $\frac{\epsilon}{2}$, in each iteration of space-narrowing sampling in Algorithm 4, if $|U| \ge (1+\frac{\epsilon}{2})(1+\frac{\epsilon}{2})z$, then

```
Algorithm 7 Distributed $(k, z)$-median/means by SNS
Input: A partition $\{P_1, P_2, ..., P_m\}$ of data set $P$ among $m$ machines, parameters $k$, $z$, $\eta$, $\epsilon$, $L$.
Output: A set $C \subseteq P$ of size at most $k$ or "No".
Round $1$ on each machine $i \in [m]$
Call the algorithm in (Chen et al., 2018) by setting $z = \frac{(1+\epsilon)z}{m}$ to obtain an $O(1)$-approximate solution $S_i$ on $P_i$ with $\frac{8(1+\epsilon)z}{m}$ outliers discarded, and denote the set of outliers discarded as $Z_i$.
$Q_i =$ SNS($Z_i$, $k$, $(1+\frac{\epsilon}{2})z$, $\eta$, $\frac{\epsilon}{2}$, $m$, $\frac{2L}{\epsilon z}$).
Assign each data point in $P_i \backslash Z_i$ to its closest center in $S_i$.
For each $s \in S_i$, let $\sigma(s)$ be the number of points assigned to $s$, and assign a weight of $\sigma(s)$ to $s$.
$S_i = S_i \cup Q_i$.
Send the weighted representation $S_i$ to the coordinator.
Round $2$ on the coordinator
$S = S_1 \cup S_2 \cup ... \cup S_m$.
For each point $p \in S$, let $w(p)$ be the weight of $p$.
$z' = (1+\epsilon)z + \sum_{p \in S} w(p) - |P|$.
if $z' < 0$ then
    Return "No".
else
    Perform a weighted $(k, t)$-median/means algorithm on $S$ with $t = z'$, and return the resulting clustering.
end if
```

$\frac{|Z_i \backslash (Z^* \cup P_L)|}{|U|} \geq \frac{(1+\frac{\epsilon}{2})(1+\frac{\epsilon}{2})z - z - \frac{\epsilon z}{2}}{(1+\frac{\epsilon}{2})(1+\frac{\epsilon}{2})z} = \frac{\frac{\epsilon}{2}}{(1+\frac{\epsilon}{2})} = \frac{\epsilon}{2+\epsilon}$. By randomly taking a sample of size $\frac{1+\frac{\epsilon}{2}}{\frac{\epsilon}{2}} \log \frac{1}{\eta} = \frac{2+\epsilon}{\epsilon} \log \frac{1}{\eta}$ from U, with probability at least $1-\eta$, we can sample at least one data point from $P_{opt} \backslash P'$ to make an uncovered optimal cluster covered with radius $\frac{4L}{\epsilon z}$. If $|U| \leq (1+\frac{\epsilon}{2})(1+\frac{\epsilon}{2})z$, we can get that in step 7 of Algorithm 4, at least one data point from $P_{opt} \backslash P'$ can still be sampled to make an uncovered optimal cluster covered. By Lemma 4.1, the total number of uncovered inliers over all machines can be bounded by $\frac{\epsilon}{2}(1+\frac{\epsilon}{2})z \leq \epsilon z$ using the fact that $\epsilon \leq 1$. By using Lemma 5.4 and putting all things together, Theorem 5.2 can be proved.

D. Complementary experimental results

D.1. Complementary Experimental Results for (k, z)-Center Problem

Table 3 shows the comparison results with fixed z and m using ours_0.99 algorithm as reference. We fix the results of ours_0.99 algorithm as 1, and give the ratios between other algorithms and ours_0.99 algorithm. We take the computation of clustering cost as an example to illustrate how to get the comparison results. For each dataset i, assume that we compare algorithm A with ours_0.99 algorithm. The ratio between algorithm A and ours_0.99 (denoted by R_A^i) is $R_A^i = \frac{1}{5} \sum_{j=1}^{j=5} \frac{S_{i,j}}{O_{i,j}}$, where $S_{i,j}$ is the clustering cost returned by algorithm A with $k = 10 \times j$, and $O_{i,j}$ is the clustering cost returned by ours_0.99 with $k = 10 \times j$. The average clustering cost of algorithm A (denoted as R_A^{avg}) is obtained by calculating the average values of all five datasets, i.e., $R_A^{avg} = \frac{1}{5} \sum_{i=1}^{i=5} R_A^i$, where R_A^i is the clustering cost obtained on the i-th dataset.

Table 4 shows the comparison results with fixed k and m using ours_0.99 algorithm as reference. We fix the results of ours_0.99 algorithm as 1, and give the ratios between other algorithms and ours_0.99 algorithm. We take the computation of clustering cost as an example to illustrate how to get the comparison results. For each dataset i, assume that we compare algorithm A with ours_0.99 algorithm. The ratio between algorithm A and ours_0.99 (denoted by T_A^i) is $T_A^i = \frac{1}{5} \sum_{j=7}^{j=11} \frac{S'_{i,j}}{O'_{i,j}}$, where $S'_{i,j}$ is the clustering cost returned by algorithm A with $z = 2^j$, and $O'_{i,j}$ is the clustering cost returned by ours_0.99 with $z = 2^j$. The average clustering cost of algorithm A (denoted as T_A^{avg}) is obtained by calculating the average values of all five datasets, i.e., $T_A^{avg} = \frac{1}{5} \sum_{i=1}^{i=5} T_A^i$, where T_A^i is the clustering cost obtained on the i-th dataset.

Table 3. Comparison results for distributed (k, z)-center with fixed z and m using ours_0.99 as reference

Datasets	Index	Algorithms		
		glz	dist_kzc_0.99	dist_kzc_0.1
Letter ($m = 5$, $z = 1024$)	Clustering Cost	1.0783	0.9953	0.9628
	Communication Cost	53.7438	1.0131	10.6955
	Time	20.4076	14.5418	42.2432
Skin ($m = 10$, $z = 1024$)	Clustering Cost	0.9373	1.0083	0.9267
	Communication Cost	49.7843	1.8273	10.7531
	Time	19.7599	10.0266	46.8667
Covertype ($m = 20$, $z = 1024$)	Clustering Cost	0.9602	1.0711	0.9807
	Communication Cost	47.5037	1.5676	10.2716
	Time	28.5633	4.1265	14.5009
Gas ($m = 20$, $z = 1024$)	Clustering Cost	0.8471	1.0629	0.9213
	Communication Cost	53.5668	1.9709	11.9300
	Time	61.2919	7.4533	31.2366
Higgs ($m = 50$, $z = 1024$)	Clustering Cost	1.1493	1.1123	1.0286
	Communication Cost	150.2571	3.3773	29.3654
	Time	36.1794	87.4402	197.3946
Average	Clustering Cost	0.9944	1.0500	0.9641
	Communication Cost	70.9711	1.9512	14.6031
	Time	33.2404	24.7177	66.4484

Table 4. Comparison results for distributed (k, z)-center with fixed k and m using ours_0.99 as reference

Datasets	Index	Algorithms		
		glz	dist_kzc_0.99	dist_kzc_0.1
Letter ($m = 5$, $k = 20$)	Clustering Cost	1.0729	1.0325	0.9806
	Communication Cost	42.4735	1.0673	8.4694
	Time	37.0074	16.6316	37.0381
Skin ($m = 10$, $k = 20$)	Clustering Cost	0.9224	1.0898	0.9231
	Communication Cost	41.6273	1.6791	12.0273
	Time	24.2500	10.3500	55.8778
Covertype ($m = 20$, $k = 20$)	Clustering Cost	0.9453	1.0224	0.9669
	Communication Cost	39.8061	1.6150	11.1283
	Time	27.3802	5.5003	14.0578
Gas ($m = 20$, $k = 20$)	Clustering Cost	0.9119	1.0742	0.8956
	Communication Cost	40.9110	1.9311	12.2475
	Time	45.0791	9.2592	27.1068
Higgs ($m = 50$, $k = 20$)	Clustering Cost	1.0486	0.9165	0.9083
	Communication Cost	106.9722	4.5867	35.7192
	Time	26.7273	57.7062	73.8065
Average	Clustering Cost	0.9802	1.0271	0.9349
	Communication Cost	54.3580	2.1758	15.9184
	Time	32.0888	19.8895	41.5774

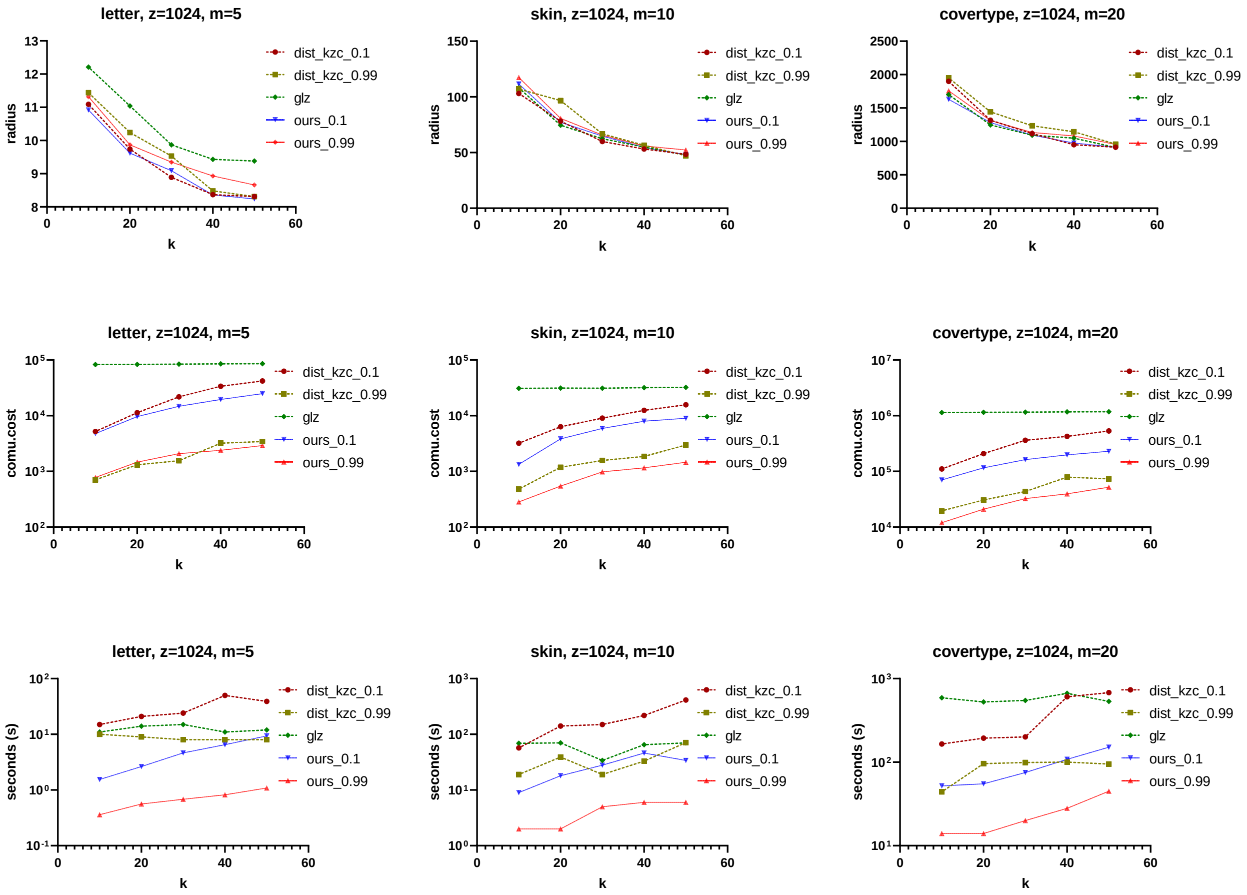

Figure 2. Comparison results of clustering performance on small datasets for distributed (k, z)-center with varying k

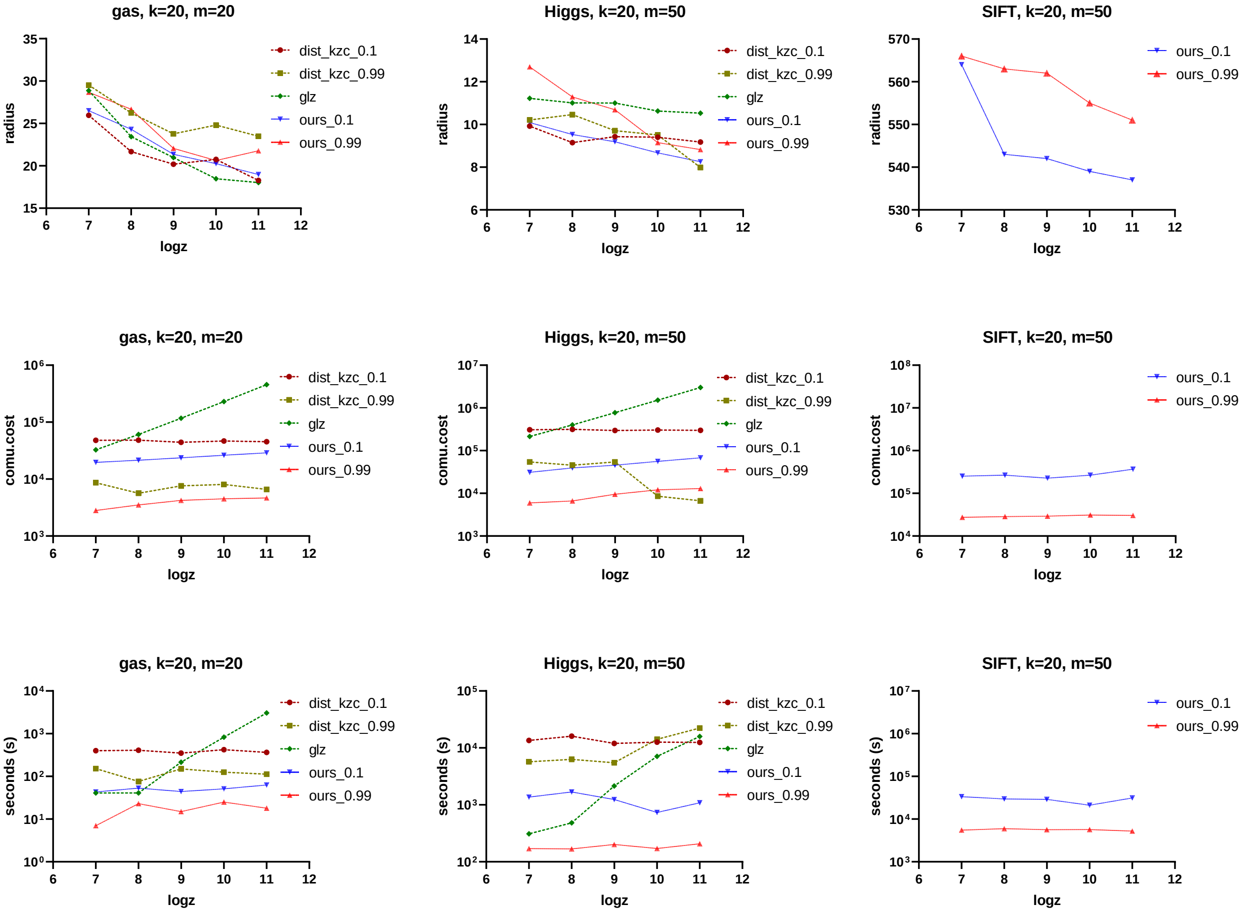

Figure 3. Comparison results of clustering performance on large datasets for distributed (k, z)-center with varying z

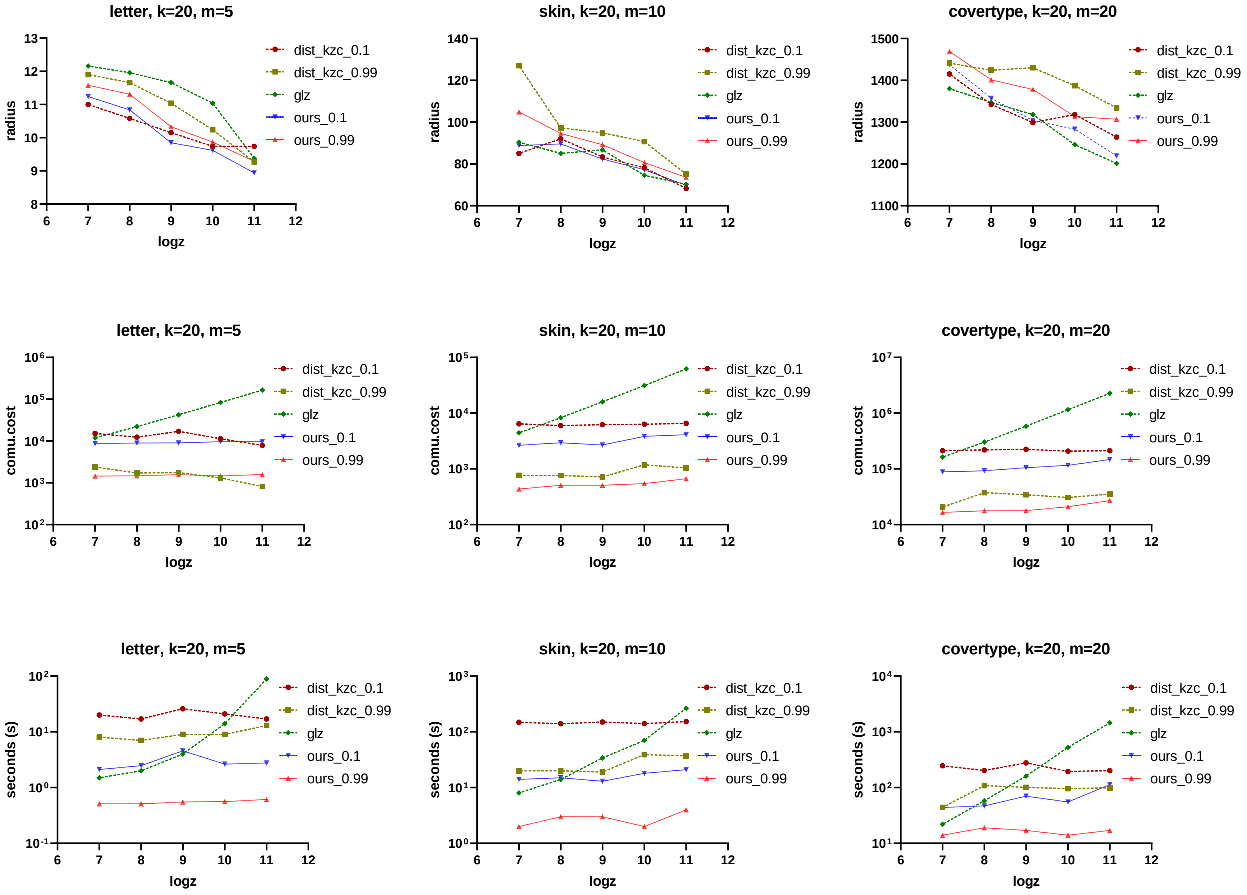

Figure 4. Comparison results of clustering performance on small datasets for distributed (k, z)-center with varying z

D.2. Experiments on Distributed (k, z)-Means

Datasets. In this section, we evaluate the performance of our distributed (k, z)-means algorithm on several real-world datasets[4] as used in (Li & Guo, 2018; Chen et al., 2018). The datasets include 3 small datasets (spambase: $4,601 \times 57$, parkinsons: $5,875 \times 16$, pendigits: $10,992 \times 16$) and 3 large datasets (KDD: $4,898,431 \times 37$, SUSY: $5,000,000 \times 18$, SIFT: $100,000,000 \times 128$).

Algorithms and parameters. In experiments, we compare our algorithm described in Algorithm 7 with other distributed algorithms. Algorithm **bel** is the one proposed by (Balcan et al., 2013), which is based on coreset construction. Algorithm **Li_0.99** is the distributed (k, z)-means algorithm proposed by (Li & Guo, 2018) with $\epsilon = 0.99$. Algorithm **ours_0.99** is our Algorithm 7 with $\epsilon = 0.99$. In our algorithm, we fix the parameter $\eta = 0.5$ and multiply the sampling rounds by a factor of $\beta = 0.01$. Algorithm **D-Sampling** is the distributed algorithm given in (Grunau & Rozhoň, 2022), which is based on D^2-sampling methods. Algorithm **Summary_Outlier** is the distributed algorithm given in (Chen et al., 2018) with $\alpha = 0.45$ and $\beta = 0.5$.

Experimental setup. Unlike the (k, z)-center problem, the furthest z data points of a (k, z)-median/means instance are unable to significantly influence the objective value. Thus, following the settings in (Chen et al., 2018; Li & Guo, 2018), we manually add the outliers using the following steps. Firstly, we normalize the dataset such that the mean and standard deviation are 0 and 1 on each dimension, respectively. Then, for each dataset, we randomly add 1% outliers that lie in range $[-\Delta, \Delta]^d$ for $\Delta = 5$. For each algorithm, the experiments are executed for five times, and we take the average results. Following the settings in (Li & Guo, 2018), the number of centers k is fixed to be 10, and we test the performance of different algorithms with varying number of outliers z, where the number of machines m is fixed to be 5 on small datasets. For large datasets, we fix the number of machines m as 20 to match the experimental setup in the (k, z)-center problem, and test the performance of different algorithms with varying number of outliers z. We compute the final clustering cost for each algorithm by removing $(1 + \epsilon)z$ outliers, same as the one used in (Li & Guo, 2018).

Results. Figure 5 shows the comparison results on small datasets with varying z, fixed m and k. It can be seen that our algorithm performs slightly better than D-Sampling method on clustering cost with faster running time. However, the communication cost of our algorithm is slightly worse than D-Sampling method. This is because, the filtering process used in our algorithm (see Summary_Outlier) has much larger communication cost compared with other algorithms. However, compared with Li_0.99 and Summary_Outlier, the communication cost of our proposed algorithm is much smaller. The coreset-based method (bel) achieves the best communication cost with the worst performance on clustering cost, since it was not designed for handling outliers.

Figure 6 shows the comparison results on large datasets with varying z, fixed m and k. For Li_0.99, it takes more than 24 hours to handle the large datasets, which indicates that the exponential running time may constraint the scalability of algorithm in (Li & Guo, 2018) for handling large-scale datasets. On large datasets, it can be seen that our proposed algorithm is much faster than D-Sampling method on datasets KDD and SUSY. On dataset SUSY, the clustering cost of our algorithm is better than D-sampling, with a slightly worse communication cost. On dataset KDD, there are no significant difference on clustering cost and communication cost between our algorithm and D-Sampling method. For running time, our algorithm is much faster than D-Sampling method on datasets KDD and SUSY.

[4] https://archive.ics.uci.edu/ml/index.php

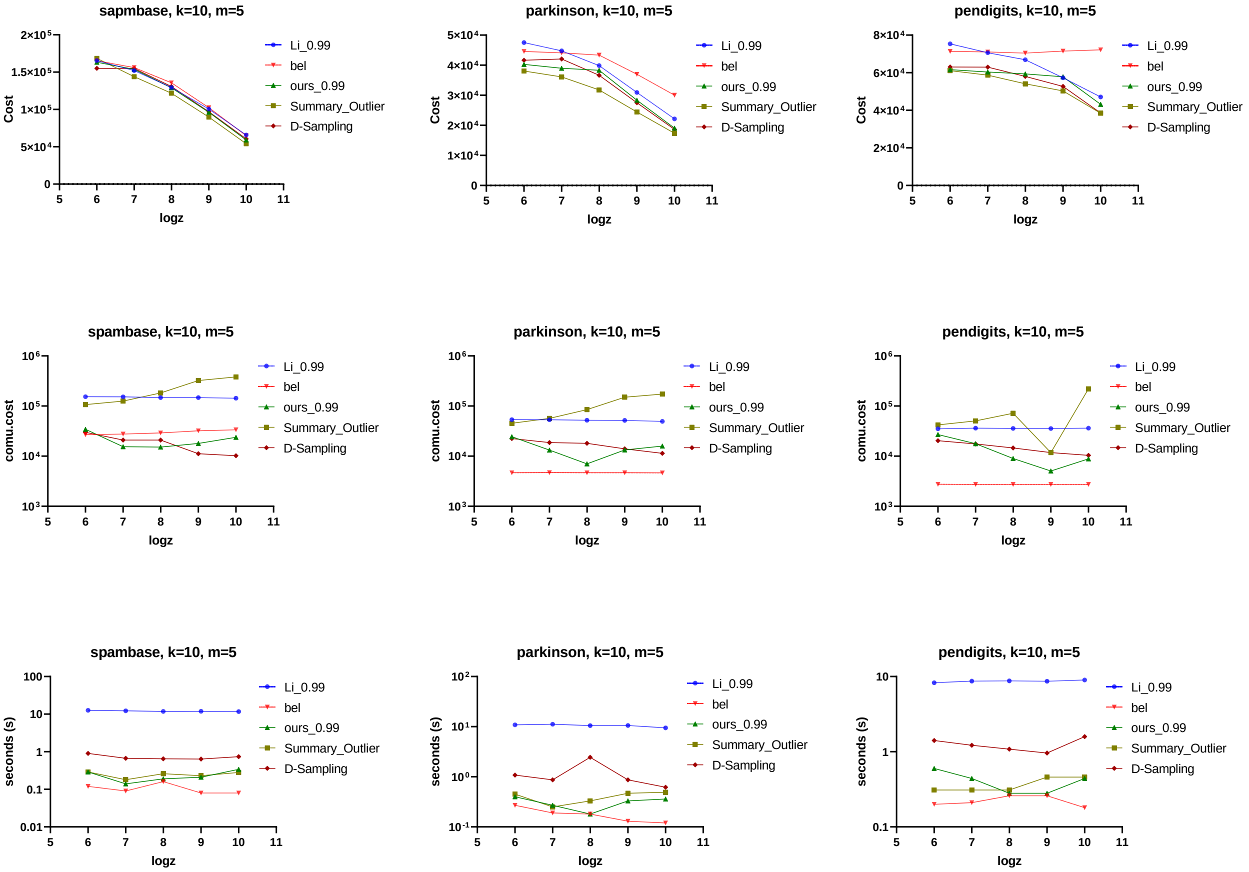

Figure 5. Comparison results of clustering performance on small datasets for distributed (k, z)-means with varying z

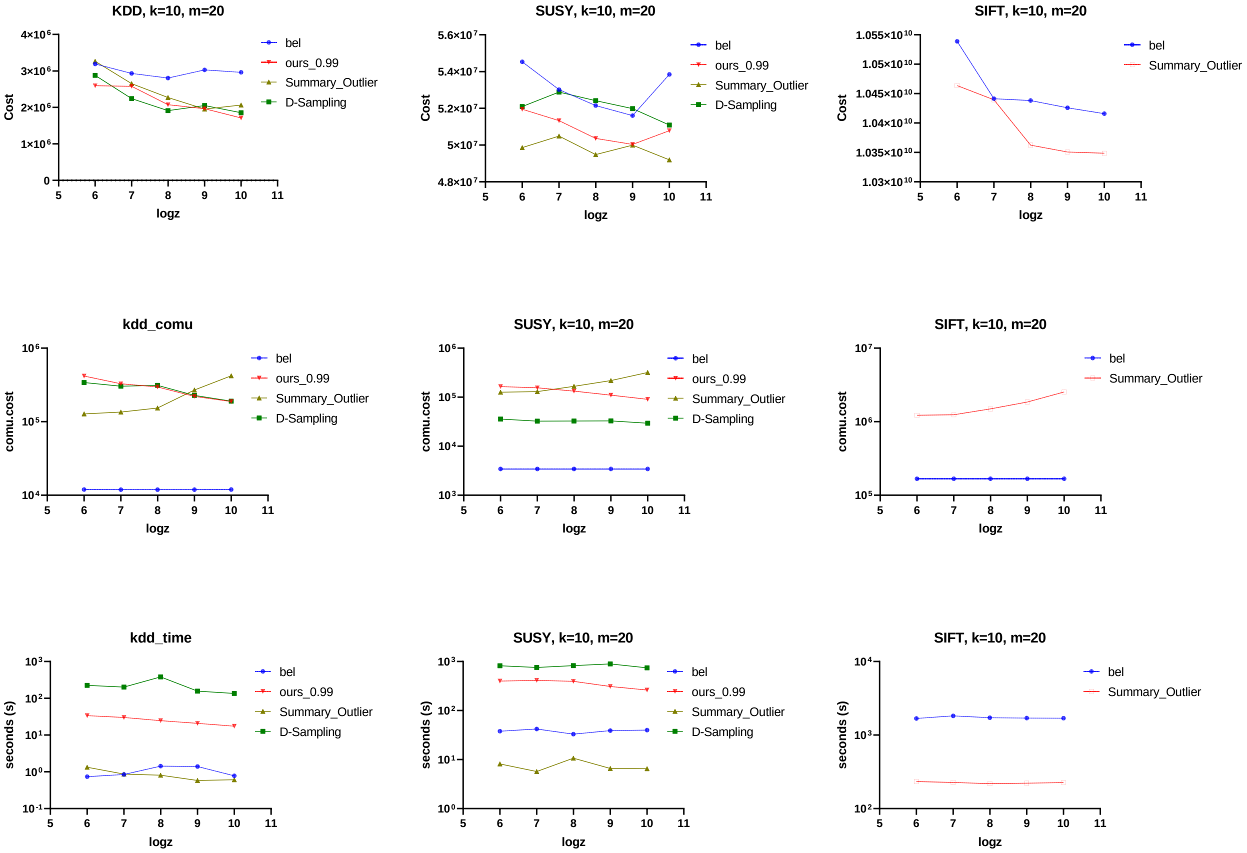

Figure 6. Comparison results of clustering performance on large datasets for distributed (k, z)-means with varying z