
Improved Guarantees for Fully Dynamic k -Center Clustering with Outliers in General Metric Spaces

Leyla Biabani

Eindhoven University of Technology
Eindhoven, The Netherlands
l.biabani@tue.nl

Annika Hennes

Heinrich Heine University Düsseldorf
Düsseldorf, Germany
annika.hennes@hhu.de

Denise La Gordt Dillie

Eindhoven University of Technology
Eindhoven, The Netherlands
lagordtdilliedenise@gmail.com

Morteza Monemizadeh

Eindhoven University of Technology
Eindhoven, The Netherlands
m.monemizadeh@tue.nl

Melanie Schmidt

Heinrich Heine University Düsseldorf
Düsseldorf, Germany
mschmidt@hhu.de

Abstract

The metric k -center clustering problem with z outliers, also known as (k, z) -center clustering, involves clustering a given point set P in a metric space (M, d) using at most k balls, minimizing the maximum ball radius while excluding up to z points from the clustering. This problem holds fundamental significance in various domains, such as machine learning, data mining, and database systems.

This paper addresses the fully dynamic version of the problem, where the point set undergoes continuous updates (insertions and deletions) over time. The objective is to maintain an approximate (k, z) -center clustering with efficient update times. We propose a novel fully dynamic algorithm that maintains a $(4 + \epsilon)$ -approximate solution to the (k, z) -center clustering problem that covers all but at most $(1 + \epsilon)z$ points at any time in the sequence with probability $1 - k/e^{\Omega(\log k)}$. The algorithm achieves an expected amortized update time of $\mathcal{O}(\epsilon^{-3}k^6 \log(k) \log(\Delta))$, and is applicable to general metric spaces. Our dynamic algorithm presents a significant improvement over the recent dynamic $(14 + \epsilon)$ -approximation algorithm by Chan, Lattanzi, Sozio, and Wang [5] for this problem.

1 Introduction

Clustering problems and algorithms play an important role across a multitude of fields, helping researchers and practitioners in the analysis of data and identification of patterns. These techniques find extensive application in diverse domains, including machine learning, where they help in categorizing and understanding complex datasets. In data mining, clustering methods are utilized to uncover hidden structures and relationships within large datasets, facilitating better decision-making and insight generation. Moreover, in image and signal processing, clustering algorithms assist in segmenting and classifying data, enabling tasks such as image recognition and signal denoising.

In bioinformatics, clustering techniques are essential for organizing biological data and identifying patterns in genetic sequences, protein structures, and gene expression profiles. Similarly, in anomaly

detection, clustering methods are employed to identify unusual or unexpected patterns in data, which may indicate potential anomalies or security breaches. Furthermore, in social network analysis, clustering algorithms help in understanding the structure and dynamics of social networks by identifying communities and influential nodes.

The k -center problem is known as one of the fundamental clustering problems. Given a set of points P in a metric space and a number k , the aim of the k -center problem is to find k centers such that the maximum distance between any point and its closest center is minimized. This can also be equivalently formulated as finding a minimum radius r and centers c_1, \dots, c_k such that the balls $\cup_{i=1}^k \mathcal{B}_P(c_i, r)$ cover the point set. The k -center problem can be 2-approximated, and this is the best possible approximation guarantee [13]. In the last decade, the focus has shifted to analyzing the problem under various complications that arise in applications.

One line of research is to study the k -center problem (and other clustering problems) in different computational models like *streaming* or for *dynamic* point sets. In the first case, points arrive sequentially, and only a summary can be stored in memory. In the second case, the point set is maintained by insertion queries and deletion queries for single points, and algorithms have to update their solution after any such query.

Another line of research is to study clustering under constraints. For example, *capacitated* clustering is very popular, i.e., limiting the number of points per cluster. However, lower bounds on cluster sizes have also been studied in the context of anonymity, and newer works have also considered constraints that model societal concerns like fair or diverse composition of clusters. In this paper, we study clustering *with outliers*. Formulated as a constraint, the k -center problem with outliers allows $k + z$ centers, but z of these have to be singletons, meaning no point may be assigned to them. We can intuitively formulate it with balls as follows: The k -center problem with outliers asks to find a minimum r and k centers c_1, \dots, c_k such that the balls $\cup_{i=1}^k \mathcal{B}_P(c_i, r)$ cover all but z points.

Solving clustering problems in the presence of outliers is a crucial task due to the common occurrence of measurement errors or other sources of significant deviation from the rest of the data in real-world datasets. Ignoring outliers can severely distort the results of clustering algorithms, leading to inaccurate groupings. To address this challenge, a common approach is to solve a clustering problem while excluding up to z data points considered as outliers.

The first approximation algorithm for the k -center problem with outliers is due to Charikar et al. [8]. The challenge when designing approximation algorithms in the presence of outliers is that one needs to show that *enough* points are covered by balls of bounded sizes around the approximate centers. It is not necessary to identify the outliers of an optimal solution *exactly* as long as the number of uncovered points remains small enough. Due to this, [8] and follow-up papers use *charging* arguments. Points covered by the solution of the algorithm are mapped to points in optimum clusters, and it is then shown that the number of uncharged points is small enough.

We explore the k -center clustering problem with z outliers within the fully dynamic model, where the point set experiences continuous updates through insertions and deletions over time. Quite some work on this problem has been done for inputs from metric spaces with *bounded doubling dimension* [2, 3, 19]. This setting allows for geometric data structures that allow easier navigation through the data set and also bounding of the number of changes that can occur due to a query by dimension-dependent volume arguments.

We study the general metric setting. General metric spaces represent a key objective for clustering algorithms due to their broad range of distance functions, ensuring applicability to any data type. Chan et al. recently showed that in general metric spaces, any dynamic $\mathcal{O}(1)$ -approximation algorithm for k -center clustering excluding at most z outliers has an amortized update time of $\Omega(z)$ [5]. For real-world applications, the fraction of outliers of the data could be arbitrarily large. Therefore, we allow for $(1 + \epsilon)z$ outliers to be excluded to avoid the dependence on z of the update time.

The only previous work in this setting is the algorithm by Chan et al. [5], which returns a solution with at most $(1 + \epsilon)z$ outliers with probability at least $1 - \delta$, for $0 < \delta \leq \frac{1}{k}$ and $\epsilon > 0$. It works by maintaining a clustering with $t := k \cdot \lceil \log_{1+\epsilon} \frac{k}{\delta} \rceil$ clusters of radius $2r$ and using this to derive a final clustering with k clusters and radius $14r$. They maintain such a clustering for all $r \in \Gamma := \{(1 + \tau)^i : d_{\min} \leq (1 + \tau)^i \leq (1 + \tau) \cdot d_{\max}, i \in \mathbb{N}\}$, with d_{\min} and d_{\max} the minimum and maximum distances, respectively, between any two points ever inserted. It is then shown that there exists an instance $r \in \Gamma$ that will approximate the optimal radius to within a factor $(14 + \tau)$,

while allowing for $(1 + \epsilon)z$ outliers, with probability at least $1 - \delta$. The amortized time per update is $\mathcal{O}(|\Gamma| \cdot \frac{k^2}{\epsilon^2} \log^2 \frac{1}{\delta})$, with $|\Gamma| = (\log \frac{d_{\max}}{d_{\min}}) / \tau$. The total memory requirement is $\mathcal{O}(|\Gamma| \cdot |n|)$ where n is the number of points in the current set.

1.1 Our contribution

We introduce a novel fully dynamic $(4 + \epsilon)$ -approximation algorithm designed to maintain a k -center clustering while allowing for at most $(1 + \epsilon)z$ outliers at any time in the sequence.

The expected amortized update time is $\mathcal{O}(\epsilon^{-3} k^6 \log(k) \log(\Delta))$ per operation (insertion or deletion). This is independent of z and of the number of points currently present in the sequence. This characteristic makes the algorithm applicable to real-world scenarios.

Notice that our data structure is continually storing an actual solution, so we can produce this solution at any time and do not need to specify additional query times (like, for example [9, 2, 18]).

Our main technical contribution is a novel combination of the sampling-based level data structure by Chan et al. [5] and the original greedy strategy by Charikar et al. [8] that enables us to achieve a much improved approximation guarantee compared to [5].

Theorem 1.1. *Let (M, d) be a metric space and $\epsilon > 0$ be an accuracy parameter. The spread ratio $\Delta = \frac{d_{\max}}{d_{\min}}$ of all points ever inserted is assumed to be bounded. There exists a randomized fully dynamic algorithm that maintains a k -center solution that allows up to $(1 + \epsilon)z$ many outliers on the current set of points. At every point in time, the current clustering is a $(4 + \epsilon)$ -approximation to an optimal solution for the (k, z) -center problem with high probability. Upon insertion or deletion of a point, the data structure is updated in amortized update time $\mathcal{O}(\epsilon^{-3} k^6 \log(k) \log(\Delta))$.*

To achieve this, we make use of a data structure that is described in Section 2.1 and maintained by the respective algorithms handling updates to the point set. The algorithm handling insertions is specified in Section 2.2, and the deletion algorithm is described in Section 2.3. Similar to the approach by Chan et al. [5], we maintain a hierarchical structure consisting of levels, each containing one cluster. Unlike their work, however, we only maintain $\leq k$ levels. For the correct radius guess, the union of these levels directly gives the desired $(4 + \epsilon)$ -approximation. If a level violates certain properties, this and all higher levels are reclustered. As opposed to [5], we do not need to recluster every time a center gets deleted but only when a cluster does not contain enough points anymore. The algorithm handling the reclustering is described in Section 2.4.

Our analysis of the approximation ratio follows a similar argument as the proof of the 3-approximation for static k -center with z outliers as given by Charikar et al. [8] and works by charging points from chosen clusters to points in optimal clusters. We follow the same iterative charging method but extend their approach by maintaining a set of artificial outliers in order to adjust the argument to our algorithm specifically.

Note that we assume that a center can be any point from the underlying metric space, if the point was present in the data set once. Requiring that centers can only be placed at currently active points, we can achieve a 6-approximation. We formalize this in Lemma G.1.

1.2 Further Related work

The k -center problem is very well understood; two 2-approximation algorithms for it are known [13, 15] and a matching lower bound is also known [16]. The k -center problem with outliers was first studied and 3-approximated in [8]. In 2016, [4] gave a 2-approximation for this problem, but the algorithm is rather complex and less amenable to practical implementation. Charikar et al. [7] developed an elegant algorithm to maintain an 8-approximation for the vanilla k -center problem in the streaming model. Streaming can be seen as a dynamic insertion-only model. McCutchen and Khuller [17] improved this algorithm to a $(2 + \epsilon)$ -approximation and also extended it to a $(4 + \epsilon)$ -approximation for k -center with outliers. The k -center problem with outliers has also been studied in the sliding window model [10, 18] that also bears some similarity with our setting. For bounded doubling dimension, [18] gave a $(3 + \epsilon)$ -approximation, but the algorithm for the general metric case is only stated as a $\mathcal{O}(1)$ -approximation.

The fully dynamic k -center problem without outliers in general metrics was studied by [1, 6, 11], and the best-known result is a $(2+\epsilon)$ -approximation with an amortized update time of $\mathcal{O}(k \text{ polylog}(n, \Delta))$ by Bateni et al. [1]. The same paper also shows that any algorithm that provably satisfies a non-trivial approximation guarantee needs $\Omega(nk)$ queries to the distance function, i.e., the amortized update time is in $\Omega(k)$, making their algorithm close to tight. The fully dynamic k -center problem without outliers was also studied for metrics with bounded doubling dimension [14, 19, 12] and in Euclidean space [20].

1.3 Preliminaries

In the preliminaries section, we provide an introduction to the (k, z) -center problem and discuss the dynamic model used in our paper.

We study the k -center clustering problem with z outliers, formally defined as follows.

Definition 1.2 ((k, z) -center clustering). Let P be a point set in a metric space (M, d) and let $k, z \in \mathbb{N}$ be two parameters. The goal is to compute a set $C \subseteq M$ of size at most k , such that the maximum distance of all but at most z points to their nearest $c \in C$ is minimized. That is, find C such that $\min_{Z \subseteq P, |Z| \leq z} \max_{x \in P \setminus Z} \min_{c \in C} d(x, c)$, with $|C| \leq k$.

We define d_{\min} and d_{\max} as the minimum and maximum distances, respectively, between any two points ever inserted. The ratio $\Delta = \frac{d_{\max}}{d_{\min}}$ is referred to as the spread ratio, and is assumed to be bounded. Let r_{OPT} be the optimal radius for the (k, z) -center clustering problem of a point set P . We also define the ball $\mathcal{B}_P(c, r) = \{p \in P : d(c, p) \leq r\}$ to be the set of points in P that are within distance r from c . If P is clear from the context, we may drop P from the definition $\mathcal{B}_P(c, r)$. For a non-negative integer m , we denote $\{1, \dots, m\}$ by $[m]$.

In the version of the (k, z) -center clustering problem that we consider, we allow $(1 + \epsilon)z$ outliers instead of strictly z . Furthermore, we require centers of clusters to be in the metric space (M, d) , but they need not be currently present in P . This is referred to as the non-discrete version of the problem. Lemma G.1 in the Appendix shows how our data structure can also support the discrete version of the problem and provides a 6-approximation solution for it.

Fully dynamic model. We consider the (k, z) -center clustering problem in the *fully dynamic model* against an *adaptive adversary*. In this model, we start with an empty point set, $P = \emptyset$, and process a sequence of operations determined by the adversary. We assume the adversary does not know the random bits chosen by our algorithm; however, it can observe the algorithm's output and adapt its responses in real time (unlike an *oblivious adversary*, which fixes a sequence of operations in advance). Each operation can be either an insertion, where a point from the metric space (M, d) is added to P , or a deletion, where a point currently in P is removed. We assume only points currently in P may be deleted. Let P^t represent the point set P after t operations. In other words, P^t consists of all points in (M, d) that have been inserted but not deleted after t operations.

2 Algorithm

To explain the main ideas of our algorithm, we start by describing an iterative offline algorithm, which gives a $(4 + \epsilon)$ approximation for the (k, z) -center clustering problem.

Offline algorithm. We start with a point set $P_1 = P$, $\epsilon > 0$ and parameters $k, z, r \in \mathbb{N}$. Here, r is a fixed guess for the optimal radius. For each iteration i , we sample a set of $S_i \subset P_i$ of $|S_i| = \psi \epsilon^{-1} k^2 \log k$ points with replacement, uniformly at random. Here, $\psi \geq 6\beta$ is a constant, where $\beta > \alpha \geq 1$ are constants. We find a point $c_i \in S_i$, such that $\mathcal{B}(c_i, 2r)$ covers at least ϕ points from P_i , where ϕ is some threshold to be defined later. We then create a new cluster $C_i = \mathcal{B}_{P_i}(c_i, 4r)$, let $P_{i+1} = P_i \setminus C_i$, and continue to the next iteration. We stop once k clusters have been created or $P_i = \emptyset$. Let λ be a random variable representing the number of iterations we complete, where λ can be at most k . After we have done λ iterations we have computed clusters $C_1, C_2, \dots, C_\lambda$ with corresponding centers $c_1, c_2, \dots, c_\lambda$. The points that are not covered by the union of these clusters will be the set of outliers that our algorithm reports. If the set of outliers is at most $(1 + \epsilon)z$ points, we can report a solution for the (k, z) -center clustering problem. This offline algorithm will be used as a

sub-routine for the fully dynamic algorithm, which will be explained in Section 2.4. The definition of ϕ and the pseudocode of the algorithm will also be given in that section.

Guesses for unknown r_{OPT} . Since the optimal r is usually not known, we can run the algorithm for all $r \in \mathcal{R} = \{(1 + \epsilon)^i : d_{\min} \leq (1 + \epsilon)^i \leq (1 + \epsilon) \cdot d_{\max}, i \in \mathbb{N}\}$. We then choose the smallest $r \in \mathcal{R}$ such that all but at most $(1 + \epsilon)z$ points are covered. We will show that this gives a $(4 + \epsilon)$ -approximation.

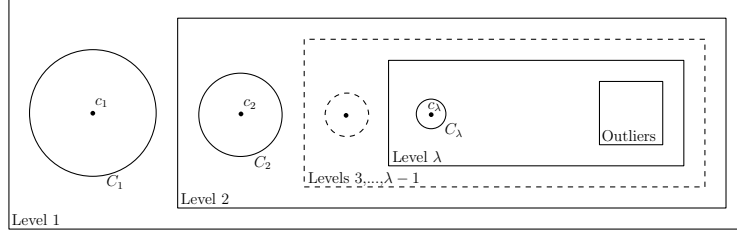


Figure 1: The λ levels constructed by our offline algorithm.

Leveling. We can visualize the output of this algorithm as (at most) $\lambda + 1$ levels, where the first $\lambda \leq k$ levels each represent a cluster, and the last level represents the outliers (See Figure 1). More precisely, level i represents cluster C_i with center c_i , which was created in the i^{th} iteration. Note that the construction of level i only considers the points in P_i , which does not include points in the lower levels $\{1, 2, \dots, i - 1\}$.

The main idea for the fully dynamic algorithm is to maintain each level i for $i \in [\lambda + 1]$ under an arbitrary number of insertions and deletions by updating the cluster C_i when necessary. In the rest of the paper, we use subscript i to refer to the level and superscript t to refer to the time. For example, P_i^t refers to the points present in level i at time t , and r_{OPT}^t refers to the optimal radius at time t . If the level or time is clear from the context, these may be omitted. For instance, if we define a situation at time t' , we will not repeat the superscript t' on every term.

2.1 Data structure and invariants

We will construct a clustering C_1, \dots, C_λ of the current set of points P^t that we store in the data structure \mathcal{D}_r introduced by [6]. If t is clear from the context, we will just write P for P^t . As we do not know the current optimal radius, we will maintain one data structure \mathcal{D}_r for every choice of $r \in \mathcal{R} = \{(1 + \epsilon)^i : d_{\min} \leq (1 + \epsilon)^i \leq d_{\max}, i \in \mathbb{N}\}$. Every data structure consists of up to $\lambda \leq k$ cluster-levels, with each level i containing a cluster C_i . In level i , we keep track of the set of the points P_i currently not covered, i. e., $P_i = P \setminus \bigcup_{j < i} C_j$. Level $\lambda + 1$ contains the outliers, i. e., $\mathcal{Z}_r = P \setminus \bigcup_{i \leq \lambda} C_i$. We denote $n_i = |P_i|$. By $\mathcal{B}(p, r)$ we denote the ball with radius r centered at p , and $\mathcal{B}_A(p, r) = \mathcal{B}(p, r) \cap A$ for $A \subseteq P$.

Let $\alpha \geq 1$. The data structure \mathcal{D}_r consists of the following components:

- ① A list $\mathcal{F}_r = \{c_1, c_2, \dots, c_\lambda\}$ of $\lambda \leq k$ centers.
- ② A list $\mathcal{L}_r = \{C_1, C_2, \dots, C_\lambda\}$ of clusters with $C_i = \mathcal{B}_{P_i}(c_i, 4r)$.
- ③ A set $\mathcal{Z}_r = P \setminus (\bigcup_{i \in [\lambda]} C_i)$ of outliers such that for all $i \in [\lambda]$ and $x \in \mathcal{Z}_r$, $d(x, c_i) > 4r$.

Let $\alpha \geq 1$ be a constant. We have the following invariants for \mathcal{D}_r .

- ① **Level invariant:** for all i , $P_{i+1} = P_i \setminus C_i$, with $C_i = \mathcal{B}_{P_i}(c_i, 4r)$.
- ② **Dense invariant:** for all $c_i \in \mathcal{F}_r$, $|\mathcal{B}_{P_i}(c_i, 2r)| \geq \min\left(z + 1, \frac{n_i - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}\right)$.

2.2 Insertion

When a new point p is inserted at time t , Procedure 3 in the appendix is executed. As input, we have \mathcal{D}_r at time t , $p \in (M, d)$, $\epsilon > 0$ and $z, k \in \mathbb{N}$. First, we check whether p is inside one of the existing clusters, in which case we can add p to such a cluster. Otherwise, we add p to \mathcal{Z}_r . Next, we check if the dense invariant is maintained after the insertion of the new point. Generally, the dense invariant can be broken in two ways:

- ❶ If the insertion of a new point results in more than $(1 + \epsilon)z$ outliers, then Lemma 3.7 shows that the dense invariant is no longer valid at some level i .
- ❷ If the addition of a point p to a cluster C_i causes the dense invariant to be violated at some lower level $j < i$, that occurs because $n_j^{t+1} = n_j^t + 1$.

If the dense invariant is broken for some level i , we recluster levels i, \dots, λ with $\lambda \leq k$ by invoking Procedure 5 as a sub-routine. Observe that if there are multiple levels i where the dense invariant is broken, we choose the lowest one.

2.3 Deletion

For the deletion of a point p at time t , Procedure 4 in the appendix is executed. The input is \mathcal{D}_r at time t , $p \in (M, d)$, $\epsilon > 0$ and $z, k \in \mathbb{N}$. First we check if p is an outlier, in which case we remove p from \mathcal{Z}_r . If p is either a center or a point in a cluster, we find cluster C_i which contains p . If cluster $C_i = \mathcal{B}(c_i, 4r)$ contains at least $\min\left(z + 1, \frac{|P_i| - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}\right)$ points, we do not re-cluster and simply remove p from C_i . Note that the underlying point set exists in the metric space (M, d) . If the center c_i of a cluster C_i is deleted, provided that the dense invariant remains valid, we can continue to utilize c_i as the center of cluster C_i , given our knowledge that the point c_i is located within the metric space (M, d) . In Lemma G.1, we explain how we can still obtain a 6-approximation if centers have to come from the current point set. If $|C_i| < \min\left(z + 1, \frac{|P_i| - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}\right)$ after the deletion of p , then it also follows that $|\mathcal{B}(c_i, 2r) \cap P_i| < \min\left(z + 1, \frac{|P_i| - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}\right)$, i.e., the dense invariant is violated on this level. In this case, we want to redistribute the points in P_i such that the levels i, \dots, λ fulfill the dense invariant. Deletion of a point in level i does not violate the invariants at levels $1, \dots, i - 1$. We re-cluster the points in $(\cup_{i \leq j \leq k} C_j) \cup \mathcal{Z}_r$ using Procedure 5. We finish by updating \mathcal{D}_r .

2.4 Clustering sub-routine

The clustering sub-routine is the offline algorithm that was described at the beginning of Section 2. The pseudocode of this sub-routine is shown in Procedure 5. We use it to iteratively build the levels of data structure \mathcal{D}_r . Two cases are distinguished based on whether z is small or large compared to the number of points in level i . In line 4, ψ is a constant with $\psi \geq 6\beta$, where $\beta > \alpha$, and α is the constant used in the dense invariant. The threshold ϕ that was introduced in Section 2 is different depending on the size of z compared to n_i . If $z + 1 \leq \frac{n_i - z}{4(k - i + 1)}$, then $\phi = \frac{n_i - z}{2(k - i + 1)}$ and if $z + 1 > \frac{n_i - z}{4(k - i + 1)}$, then $\phi = \frac{n_i - z}{k - i + 1} - \frac{\epsilon z}{\beta k}$. If we find multiple points p^* in line 6 or 12 we choose one arbitrarily.

2.5 All aspects combined

The first step will be to initialize all \mathcal{D}_r such that $\mathcal{F}_r, \mathcal{L}_r, \mathcal{Z}_r = \emptyset$. Then, the algorithm waits for an insertion or deletion operation. If a point is inserted, Procedure 3 is executed, and if a point is deleted, Procedure 4 is executed. The algorithm is ran simultaneously for all $r \in \mathcal{R} = \{(1 + \epsilon)^i : d_{\min} \leq (1 + \epsilon)^i \leq (1 + \epsilon) \cdot d_{\max}, i \in \mathbb{N}\}$.

In the next sections, it will be shown that a $(4 + \epsilon)$ -approximation is given by the clustering \mathcal{L}_r with $r \in \mathcal{R}$ the smallest r for which $|\mathcal{Z}_r| \leq (1 + \epsilon)z$.

3 Analysis

We begin our analysis of the dynamic algorithm by introducing the concept of a dense cluster and specifying the criteria for a valid solution.

Definition 3.1 (Dense cluster). A cluster $\mathcal{B}_{P_i}(c_i, 4r)$ is dense with respect to point set P_i if it satisfies the dense invariant. That is, $|\mathcal{B}_{P_i}(c_i, 2r)| \geq \min\left(z + 1, \frac{n_i - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}\right)$.

Observe that for the dense invariant, we consider the ball $\mathcal{B}_{P_i}(c_i, 2r)$ with a radius of $2r$. However, a cluster (as seen in line 18 of Procedure5) corresponds to the points within the ball $\mathcal{B}_{P_i}(c_i, 4r)$, which has a radius of $4r$.

Definition 3.2 (Valid solution). A solution $\mathcal{B}_{P_1}(c_1, 4r) \cup \dots \cup \mathcal{B}_{P_\lambda}(c_\lambda, 4r)$ with $\lambda \leq k$ for point set P_1 is valid if it covers all but at most $(1 + \epsilon)z$ points from P_1 . We refer to each cluster $\mathcal{B}_{P_j}(c_j, 4r)$ for $j \in \{1, \dots, \lambda\}$ of a valid solution as a valid cluster.

Lemma 3.3 (Running time offline algorithm). *The running time of the offline algorithm, shown in Procedure 5, is $\mathcal{O}(n\epsilon^{-1}k^3 \log(k))$, where $n = |P|$.*

Proof. In each iteration of the while-loop, we need to compute $|\mathcal{B}(p, 2r)|$ for all $p \in S_i$. Since $|S_i| = \mathcal{O}(\epsilon^{-1}k^2 \log k)$, this takes $\mathcal{O}(n\epsilon^{-1}k^2 \log k)$ time. There can be at most k iterations of the while-loop and hence the total running time is $\mathcal{O}(n\epsilon^{-1}k^3 \log k)$. \square

3.1 Maintaining invariants

In Supplementary Section B, we prove the following three lemmas. Lemmas 3.4 and 3.5 show that the dense and level invariants are maintained during the insertion or deletion of a point, respectively. Additionally, Lemma 3.6 shows that both invariants are maintained when invoking Procedure 5 as a subroutine upon the insertion or deletion of an arbitrary point with high probability if $r \geq r_{\text{OPT}}$. The case where $r < r_{\text{OPT}}$ is considered in Lemmas E.1 and E.2.

Lemma 3.4 (Procedure 3 maintains invariants). *Assume that at time t , we have point set P^t , data structure $\mathcal{D}_r = (\mathcal{F}_r, \mathcal{L}_r, \mathcal{Z}_r)$. We assume that the level and dense invariants hold at time t and $r \geq r_{\text{OPT}}^{t+1}$. At the start of time $t + 1$, we insert point p using Procedure 3. After the insertion, the level and dense invariants still hold with probability 1 if Procedure 5 was not called and with the probability of at least $1 - \frac{2(k-i+1)}{e^{\Psi \log k}}$, where $\Psi \geq 1$ if Procedure 5 was not called.*

Lemma 3.5 (Procedure 4 maintains invariants). *Assume that at time t , we have point set P^t , instance $\mathcal{D}_r = (\mathcal{F}_r, \mathcal{L}_r, \mathcal{Z}_r)$, parameters $k, z \in \mathbb{N}$ and $\epsilon > 0$. We assume that the level and dense invariants hold at time t and $r \geq r_{\text{OPT}}^{t+1}$. At the start of time $t + 1$, we delete an arbitrary point p using Procedure 4. After the deletion, the level and dense invariants hold with probability 1 if Procedure 5 was not called, and with probability $1 - \frac{2(k-i+1)}{e^{\Psi \log k}}$ with $\Psi \geq 1$ if Procedure 5 was called.*

Lemma 3.6 (Procedure 5 maintains invariants with high probability). *Suppose the level and dense invariants hold for all levels $j < i$ and we call Procedure 5 on P_i as the result of an insertion or deletion. Let $\lambda \leq k$ be a random variable representing the number of levels we have after completing Procedure 5. If $r \geq r_{\text{OPT}}$, Procedure 5 maintains the level and dense invariants for all levels j with $i \leq j \leq \lambda$ with probability at least $1 - \frac{2(k-i+1)}{e^{\Psi \log k}}$, with $\Psi \geq 1$.*

3.2 Approximation guarantee

Next, we prove that if the invariants hold, our data structure $\mathcal{D}_r = (\mathcal{F}_r, \mathcal{L}_r, \mathcal{Z}_r)$ contains a valid solution to the k -center with z outliers problem. Furthermore, if $r < (1 + \epsilon)r_{\text{OPT}}$, \mathcal{D}_r provides a $(4 + \epsilon)$ -approximation. Without loss of generality, for simplicity we assume that \mathcal{D}_r contains k levels. The proof of Lemma 3.7 follows the structure of the proof given in [8]. This proof uses the greediness of the algorithm to argue that the balls in the solution cover enough points to charge to. Here, we have to modify the proof to work with the dense invariant.

Lemma 3.7 (Approximation guarantee). *Let P be the current point set. Assume that the level invariant and the dense invariant hold for all levels $i \leq k$. Then we have the following guarantees:*

- ① **Valid solution:** *If $r_{\text{OPT}} \leq r$, then $\mathcal{B}(c_1, 4r) \cup \dots \cup \mathcal{B}(c_k, 4r)$ covers all but at most $(1 + \frac{\epsilon}{\alpha})z$ outliers in P .*
- ② **Approximate solution:** *If $r_{\text{OPT}} \leq r < (1 + \epsilon)r_{\text{OPT}}$, then $\mathcal{B}(c_1, 4r) \cup \dots \cup \mathcal{B}(c_k, 4r)$ gives a $(4 + \epsilon)$ -approximation for the k -center with z outliers problem on P .*

Proof. Recall that the level invariant states that for all i , we have $P_{i+1} = P_i \setminus \mathcal{B}(c_i, 4r)$ and the dense invariant states that for all i , we have that $|\mathcal{B}(c_i, 2r) \cap P_i| \geq \min(z + 1, \frac{nz-z}{k-i+1} - \frac{\epsilon z}{\alpha k})$.

Assume that in the optimal solution, we have balls O_1, \dots, O_k with radius $\leq r_{\text{OPT}}$. The union of these balls covers all but z points in P . In order to prove Part ①, we aim to charge all but at most $\frac{\epsilon}{\alpha}z$ points of the optimal solution to points in our solution $\mathcal{B}(c_1, 4r) \cup \dots \cup \mathcal{B}(c_k, 4r)$. We prove this by induction, and Part ② will then follow easily. In order to construct the charging argument, we need to argue that there are enough points in our solution to charge the points in the optimal balls. To this end, we construct modified optimal balls O'_1, \dots, O'_k , where $O'_i \subseteq O_i$ for every $i \leq k$.

For the base case, we have $O'_1 = O_1, \dots, O'_k = O_k$. We will show that we can order the modified balls in such a way, that at the end of time step i , all but at most $\frac{\epsilon z}{\alpha k} \cdot i$ points from the first i modified balls are charged to distinct points in $\mathcal{B}(c_1, 4r) \cup \dots \cup \mathcal{B}(c_i, 4r)$. This will allow us to prove that our solution covers at least as many points as the optimal solution.

Assume that all but at most $\frac{\epsilon z}{\alpha k} \cdot (i-1)$ points in the first $i-1$ modified balls $O'_1 \cup O'_2 \cup \dots \cup O'_{i-1}$ have been charged to distinct points in $\mathcal{B}(c_1, 4r) \cup \dots \cup \mathcal{B}(c_{i-1}, 4r)$ and consider iteration i . We distinguish two cases, namely if $\mathcal{B}(c_1, 2r) \cup \dots \cup \mathcal{B}(c_i, 2r)$ intersects one of the remaining modified balls, or if it does not. The charging argument for each case proceeds as follows:

Case 1. Case 1 is when $\mathcal{B}(c_1, 2r) \cup \dots \cup \mathcal{B}(c_i, 2r)$ intersects a remaining modified ball, call this ball O'_i . Note that O'_i will be covered entirely by $\mathcal{B}(c_1, 4r) \cup \dots \cup \mathcal{B}(c_i, 4r)$, since $r \geq r_{\text{OPT}}$. Hence, we charge the points of O'_i to themselves and mark these points as covered. (See case 1 in Figure 2.) We call this charging *rule I*. Since the modified balls are disjoint¹, any point can be charged only once (to itself) by this rule. Next, we update O'_1, O'_2, \dots, O'_k .

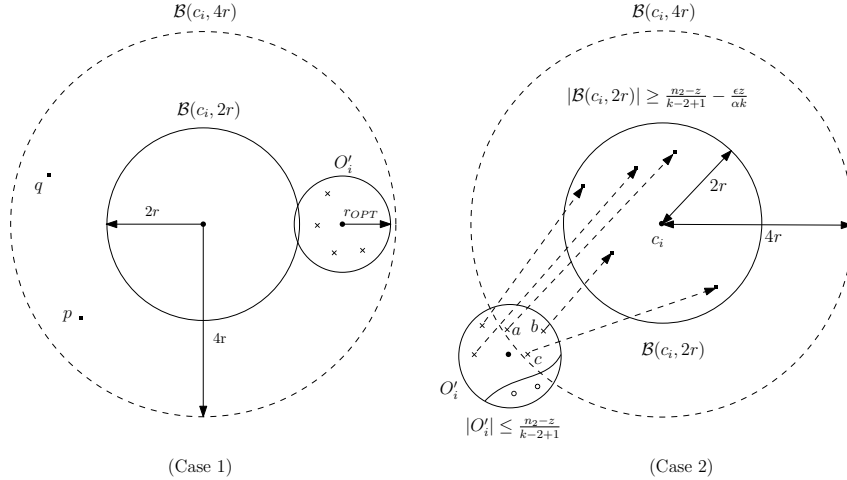


Figure 2: Visualization of *charging rules I and II*. For case 1, we see that $\mathcal{B}(c_i, 2r)$ and O_i intersect. As a result, $\mathcal{B}(c_i, 4r)$ covers all points in O_i . Points p, q are in set Z_i^c . For case 2, the balls $\mathcal{B}(c_1, 2r)$ and $\mathcal{B}(c_2, 2r)$ do not intersect the optimal cluster O'_i . The crossed points in O'_2 are charged to black squared in $\mathcal{B}(c_2, 2r)$. The circle points in O'_2 are not charged to any point. Points a, b, c are in Z_i^d .

We maintain two sets of credit points that we save and may use for future charging purposes. First, the set $Z_i^c = \mathcal{B}(c_i, 4r) \setminus (O'_1 \cup \dots \cup O'_k)$ which is the set of points covered by $\mathcal{B}(c_i, 4r)$ that are not covered by $O'_1 \cup O'_2 \cup \dots \cup O'_k$. We let $z_i^c = |Z_i^c|$ be the number of such points. In Figure 2, points p, q are such points. Observe that no point is charged to points in Z_i^c , allowing us to use them later. We refer to these points as *credit points*.

There may also be previous modified balls O'_j , with $j < i$ that were considered in case 2 and are still present in P_i . More specifically, let Z_i^d be the set of (there may exist) points in $O'_j \cap \mathcal{B}(c_j, 4r)$ that have been charged to distinct points in $\mathcal{B}(c_j, 4r)$ (See the discussion of case 2, below). For example, points a, b, c in case 2 of Figure 2. Let $z_i^d = |Z_i^d|$ be the number of such points. Since no points

¹In fact, original balls O_1, \dots, O_k can overlap, but we can make them disjoint by assigning any point that is inside multiple balls to just one of them arbitrarily.

are charged to points in Z_i^d , we save them as *credit points* for future charging purposes. We now update O'_1, O'_2, \dots, O'_k as follows: O'_1, \dots, O'_i stay the same and we define $z_i^c + z_i^d$ artificial outliers in $(O'_{i+1} \cup \dots \cup O'_k) \cap P_{i+1}$.

Case 2. Case 2 occurs if $\mathcal{B}(c_1, 2r) \cup \dots \cup \mathcal{B}(c_i, 2r)$ does not intersect any of the remaining modified balls. See Figure 2. Let O'_i be one of the remaining modified balls covering $\leq \frac{n_i - z}{k - i + 1}$ points. Note that for finding O'_i we do not count points that have been defined as artificial outliers, since these artificial outliers are already covered by balls of radius $4r$ in previous levels and we do not cover them by the remaining modified balls. We prove in Lemma 3.8 that such a ball O'_i exists.

Using the assumption of this lemma that the dense invariant holds, we show in Lemma 3.9 that $|\mathcal{B}(c_i, 2r)| \geq \frac{n_i - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}$. In this way, we find an upper bound for the number of points in O'_i and a lower bound for the number of points in $\mathcal{B}(c_i, 2r)$. Now, we charge all but at most $\frac{\epsilon z}{\alpha k}$ points of O'_i to the points of $\mathcal{B}(c_i, 2r)$ and mark these points as charged. We call this charging *rule II*. Note the difference between charging argument in Case 1 and 2. In Case 1, we charge the points in O'_i to themselves, but in Case 2, we charge them to points covered by $\mathcal{B}(c_i, 2r)$. Recall that $\mathcal{B}(c_i, 2r)$ and O'_i are disjoint.

Observe that points in balls O'_{i+1}, \dots, O'_k will not be charged to points in $\mathcal{B}(c_i, 2r)$. Indeed, points in balls O'_{i+1}, \dots, O'_k are charged to the balls that our algorithms find either using rule I or rule II. However, this will not happen based on rule I since $\mathcal{B}(c_i, 2r)$ is disjoint from all balls O'_{i+1}, \dots, O'_k . Moreover, rule II cannot also be applied since among balls $O'_i, O'_{i+1}, \dots, O'_k$, we already charged the points in O'_i to $\mathcal{B}(c_i, 2r)$ and points in O'_{i+1}, \dots, O'_k will be charged to points in balls $\mathcal{B}(c_{i+1}, 2r), \dots, \mathcal{B}(c_\lambda, 2r)$.

Next, we consider how we update O'_1, O'_2, \dots, O'_k . Similar to case 1, we define the size of two sets of credits points. First, the variable $z_i^c = |\mathcal{B}(c_i, 4r) \setminus (O'_1 \cup \dots \cup O'_k)|$ that corresponds to the number of points covered by $\mathcal{B}(c_i, 4r)$ that are not covered by $O'_1 \cup O'_2 \cup \dots \cup O'_k$. There are $z_i^c - |O'_i|$ points from $\mathcal{B}(c_i, 4r)$ that are free (i.e., no point is charged to) and can still be charged. We consider these points as credits that we save and may use for future charging purposes.

There may also be previous modified balls O'_j , with $j < i$ that were considered in case 2 and are still present in P_i . More specifically, let Z_i^d be the set of (there may exist) points in $O'_j \cap \mathcal{B}(c_j, 4r)$ that have been charged to distinct points in $\mathcal{B}(c_j, 4r)$. For example, points a, b, c in case 2 of Figure 2. Let $z_i^d = |Z_i^d|$ be the number of such points. Since no points are charged to points in Z_i^d , we save them as *credit points* for future charging purposes. We now update O'_1, O'_2, \dots, O'_k as follows: O'_1, O'_2, \dots, O'_i stays the same and we define $(z_i^c - |O'_i|) + z_i^d$ artificial outliers in $(O'_{i+1} \cup \dots \cup O'_k) \cap P_{i+1}$.

Now, for both cases, we apply charging rule I to any points in the remaining modified balls $O'_{i+1} \cup \dots \cup O'_k$ that are inside $\mathcal{B}(c_i, 4r)$. These points are then marked as covered. See Figure 3 for an example. Note that these points will not be in $P_{i+1} = P_i \setminus \mathcal{B}(c_i, 4r)$. We assumed that $O_1 \cup O_2 \cup \dots \cup O_k$ covers all but at most z points. After the charging has taken place and the modified clustering $O'_1 \cup O'_2 \cup \dots \cup O'_k$ has been constructed, we have that $|O'_1 \cup O'_2 \cup \dots \cup O'_k| = n - z - \sum_{j \text{ in case 1}} (z_j^c + z_j^d) - \sum_{j \text{ in case 2}} (z_j^c - |O'_j| + z_j^d)$.

Note that $|O'_j|$ refers to the number of points covered by the modified ball O'_j at the time of iteration j . Then, by the way we have charged $O'_1 \cup \dots \cup O'_k$ to our solution $\mathcal{B}(c_1, 4r) \cup \dots \cup \mathcal{B}(c_k, 4r)$, we obtain $|\mathcal{B}(c_1, 4r) \cup \dots \cup \mathcal{B}(c_k, 4r)| \geq |O'_1 \cup O'_2 \cup \dots \cup O'_k| - \frac{\epsilon z}{\alpha} + \sum_{j \text{ in case 1}} (z_j^c + z_j^d) + \sum_{j \text{ in case 2}} (z_j^c - |O'_j| + z_j^d) = n - (1 + \frac{\epsilon}{\alpha})z$. This concludes the proof of Part ① of the lemma. It follows easily that when $r_{\text{OPT}} \leq r < (1 + \epsilon)r_{\text{OPT}}$, Part ② also holds, which completes the proof of this lemma. \square

Lemma 3.8. *Let i be an iteration of the charging argument above such that we are in case 2. This means that $\mathcal{B}(c_1, 2r) \cup \dots \cup \mathcal{B}(c_i, 2r)$ does not intersect any of the remaining modified balls. Then, there must be a remaining modified ball covering $\leq \frac{n_i - z}{k - i + 1}$ points.*

For the proof of Lemma 3.8, see Supplementary Section C.

Lemma 3.9 (Coverage of $B(c_i, 2r)$). *When we are in case 2 of the charging argument for some iteration i , we must have that $|\mathcal{B}(c_i, 2r)| \geq \frac{n_i - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}$.*

For the proof of Lemma 3.9, see Supplementary Section C.

3.3 Duration of dense clusters

After running Procedure 5 as a subroutine, we know that each cluster C_i covers at least ϕ points. More specifically, for each level i , if $z + 1 \leq \frac{n_i - z}{4(k - i + 1)}$, then $|\mathcal{B}(c_i, 2r)| \geq \frac{n_i - z}{2(k - i + 1)}$ and if $z + 1 > \frac{n_i - z}{4(k - i + 1)}$, then $|\mathcal{B}(c_i, 2r)| \geq \frac{n_i - z}{k - i + 1} - \frac{\epsilon z}{\beta k}$. The following two lemmas show that when C_i satisfies one of these two constraints, it will be dense for a significant number of arbitrary operations. This will lead to the amortized update time being independent of n . The remaining lemmas are split into two cases, corresponding to the two cases in Procedure 5.

Lemma 3.10 (Duration of dense cluster for z is small). *Assume that we are currently at time t . Let us consider a level i in which $z + 1 \leq \frac{n_i^t - z}{4(k - i + 1)}$. Let $p = \arg \max_{p' \in S_i} |\mathcal{B}_{P_i}(p', 2r)|$, and $\mathcal{B}_{\max} = \mathcal{B}_{P_i}(p, 2r)$. Assume that $\frac{n_i^t - z}{2(k - i + 1)} \leq |\mathcal{B}_{\max}|$. Then, we can add $\mathcal{B}_{P_i}(p, 4r)$ as a cluster in our solution, and this cluster will be dense until time $t' = t + t^*$, with $t^* \geq \frac{n_i^t - z}{4(k - i + 1)}$.*

For the proof of Lemma 3.10, see Supplementary Section D.

Lemma 3.11 (Duration of dense cluster for z is large). *Assume that we are currently at time t . Let $z + 1 > \frac{n_i^t - z}{4(k - i + 1)}$ for some level i . Let $p = \arg \max_{p' \in S_i} |\mathcal{B}_{P_i}(p', 2r)|$, and $\mathcal{B}_{\max} = \mathcal{B}_{P_i}(p, 2r)$. Assume that $\frac{n_i^t - z}{k - i + 1} - \frac{\epsilon z}{\beta k} \leq |\mathcal{B}_{\max}|$. Then, we can add $\mathcal{B}_{P_i}(p, 4r)$ as a cluster in our solution, and this cluster will be dense until time $t' = t + t^*$, with $t^* = \Omega(\frac{\epsilon z}{k})$.*

For the proof of Lemma 3.11, see Supplementary Section D.

3.4 Small radius guesses

It has not yet been considered what will happen if Procedure 5 fails on some level i . That is, when there exists no $p \in S_i$ such that $\mathcal{B}_{P_i}(p, 2r)$ covers sufficiently many points. Lemmas E.1 and E.2 show that if Procedure 5 fails at some level, then with high probability the guess of the optimal radius for that specific instance is too small ($r < r_{\text{OPT}}$). In this case, we postpone the construction of that level to a time $t + t^*$, referred to as t'_r in the algorithm.

3.5 Computing update time

Finally, we prove the update time of our algorithm. Lemma 3.4, Lemma 3.5 and Lemma 3.6 prove that the algorithms maintain the invariants with high probability. Together with Lemma 3.7 and Lemma F.1 this implies Theorem 1.1.

3.6 Robustness to adversarial inputs

Our dynamic algorithm is robust against an adaptive adversary. Specifically, we do not assume that the adversary has predetermined the sequence of updates in advance, as with an oblivious adversary. Instead, the adversary can query the insertion and deletion updates in an online manner, with knowledge of our solution.

We only use randomization to generate the sample set S_i in Procedure 5, which is then used to construct the levels. This randomness affects only the probability of failure in this procedure. After the insertion or deletion updates, we do not need to reconstruct a level until it no longer satisfies the invariants. The guarantees we provide for how long a level can remain valid are all in the worst case; therefore, they hold even when an adaptive adversary chooses the insertion and deletion operations online.

3.7 Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful comments. Annika Hennes' and Melanie Schmidt's research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 456558332.

References

- [1] MohammadHossein Bateni, Hossein Esfandiari, Hendrik Fichtenberger, Monika Henzinger, Rajesh Jayaram, Vahab Mirrokni, and Andreas Wiese. Optimal fully dynamic k -center clustering for adaptive and oblivious adversaries. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2677–2727. Society for Industrial and Applied Mathematics, 2023.
- [2] Leyla Biabani, Annika Hennes, Morteza Monemizadeh, and Melanie Schmidt. Faster query times for fully dynamic k -center clustering with outliers. *Advances in Neural Information Processing Systems*, 36, 2024.
- [3] Matteo Ceccarello, Andrea Pietracaprina, and Geppino Pucci. Solving k -center clustering (with outliers) in mapreduce and streaming, almost as accurately as sequentially. *Proc. VLDB Endow.*, 12(7):766–778, 2019.
- [4] Deeparnab Chakrabarty, Prachi Goyal, and Ravishankar Krishnaswamy. The non-uniform k -center problem. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 55 of *LIPICs. Leibniz Int. Proc. Inform.*, pages Art. No. 67, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2016.
- [5] T-H Hubert Chan, Silvio Lattanzi, Mauro Sozio, and Bo Wang. Fully dynamic k -center clustering with outliers. *Algorithmica*, 86(1):171–193, 2024.
- [6] TH Hubert Chan, Arnaud Guerqin, and Mauro Sozio. Fully dynamic k -center clustering. In *Proceedings of the 2018 World Wide Web Conference*, pages 579–587, Lyon, France, 2018. International World Wide Web Conferences Steering Committee.
- [7] Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 626–635, 1997.
- [8] Moses Charikar, Samir Khuller, David M. Mount, and Giri Narasimhan. Algorithms for facility location problems with outliers. In S. Rao Kosaraju, editor, *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA*, pages 642–651. ACM/SIAM, 2001.
- [9] Mark de Berg, Leyla Biabani, and Morteza Monemizadeh. k -center clustering with outliers in the MPC and streaming model. *CoRR*, abs/2302.12811, 2023.
- [10] Mark de Berg, Morteza Monemizadeh, and Yu Zhong. k -center clustering with outliers in the sliding-window model. In Petra Mutzel, Rasmus Pagh, and Grzegorz Herman, editors, *29th Annual European Symposium on Algorithms, ESA 2021, September 6-8, 2021, Lisbon, Portugal (Virtual Conference)*, volume 204 of *LIPICs*, pages 13:1–13:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
- [11] Hendrik Fichtenberger, Monika Henzinger, and Andreas Wiese. On fully dynamic constant-factor approximation algorithms for clustering problems. *CoRR*, abs/2112.07217, 2021.
- [12] Jinxiang Gan and Mordecai J Golin. Fully dynamic k -center in low dimensions via approximate furthest neighbors. In *2024 Symposium on Simplicity in Algorithms (SOSA)*, pages 269–278. SIAM, 2024.
- [13] Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical computer science*, 38:293–306, 1985.

- [14] Gramoz Goranci, Monika Henzinger, Dariusz Leniowski, Christian Schulz, and Alexander Svozil. Fully dynamic k -center clustering in low dimensional metrics. In *2021 Proceedings of the Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 143–153. SIAM, 2021.
- [15] Dorit S Hochbaum and David B Shmoys. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM (JACM)*, 33(3):533–550, 1986.
- [16] Wen-Lian Hsu and George L Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979.
- [17] Richard Matthew McCutchen and Samir Khuller. Streaming algorithms for k -center clustering with outliers and with anonymity. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 165–178. Springer, 2008.
- [18] Paolo Pellizzoni, Andrea Pietracaprina, and Geppino Pucci. k -center clustering with outliers in sliding windows. *Algorithms*, 15(2):52, 2022.
- [19] Paolo Pellizzoni, Andrea Pietracaprina, and Geppino Pucci. Fully dynamic clustering and diversity maximization in doubling metrics. In *Algorithms and Data Structures Symposium*, pages 620–636. Springer, 2023.
- [20] Melanie Schmidt and Christian Sohler. Fully dynamic hierarchical diameter k -clustering and k -center. *arXiv preprint arXiv:1908.02645*, 2019.

A Missing Pseudocode

In this section, we provide the missing pseudocodes. We start by Procedure INITIALIZATION, which is called to start the algorithm. This procedure defines the set \mathcal{R} and then creates a counter for time t , and the elements of our data structure \mathcal{F}_r , \mathcal{L}_r , and \mathcal{Z}_r that we discussed before. It also creates the variable t'_r , which is used when the radius r is smaller than r_{OPT} . If $r < r_{\text{OPT}}$, we may fail to find the p^* that we desired for a level i in Procedure OFFLINECLUSTER, and then have to stop the level construction. In this case, we set t'_r to $t + t^*$ to postpone the construction of that level to time $t + t^*$. See Lemmas E.1 and E.2 for the choice of t^* . If t'_r is set to -1 , it means there is currently no construction to be postponed. We assume all the parameters and variables defined in Procedure INITIALIZATION are global, and the other procedures have access to them.

Then, we state Procedure UPDATE to handle the insertion or deletion of a point p . Depending on whether the query is an insertion or a deletion, it calls INSERT(p, r) or DELETE(p, r) for different values of r . Then it updates the counter for time. Next, if there is any level construction that was postponed to the current time t , it handles it.

Procedure 1 INITIALIZATION($k, z, \epsilon, d_{\min}, d_{\max}$)

```

1:  $t \leftarrow 0$ 
2:  $\mathcal{R} := \{(1 + \epsilon)^i : d_{\min} \leq (1 + \epsilon)^i \leq (1 + \epsilon)d_{\max}, i \in \mathbb{N}\}$ 
3: for all  $r \in \mathcal{R}$  do
4:    $\mathcal{F}_r \leftarrow \emptyset, \mathcal{L}_r \leftarrow \emptyset, \mathcal{Z}_r \leftarrow \emptyset$ 
5:    $t'_r \leftarrow -1$ 
6: end for

```

Procedure 2 UPDATE(p)

```

1: for all  $r \in \mathcal{R}$  do
2:   if UPDATE( $p$ ) is an insertion then
3:     INSERT( $p, r$ )
4:   else if UPDATE( $p$ ) is a deletion then
5:     DELETE( $p, r$ )
6:   end if
7: end for
8:  $t \leftarrow t + 1$ 
9: for all  $r \in \mathcal{R}$  do
10:  if  $t'_r = t$  then
11:     $\lambda_r \leftarrow |\mathcal{F}_r|$ 
12:    OFFLINECLUSTER( $\mathcal{Z}_r, \lambda_r + 1, r$ )
13:  end if
14: end for

```

Procedure 3 INSERT(p, r)

```

1:  $\lambda \leftarrow |\mathcal{F}_r|, \{c_1, c_2, \dots, c_\lambda\} \leftarrow \mathcal{F}_r$ , and  $\{C_1, C_2, \dots, C_\lambda\} \leftarrow \mathcal{L}_r$ 
2: if  $d(p, c_i) \leq 4r$  for some  $i \in [\lambda]$  then
3:    $C_i \leftarrow C_i \cup p$ 
4: else
5:    $\mathcal{Z}_r \leftarrow \mathcal{Z}_r \cup p$ 
6: end if
7: For each level  $\ell, P_\ell \leftarrow (\cup_{\ell \leq j \leq \lambda} C_j) \cup \mathcal{Z}_r$ 
8: for  $i = 1$  to  $\lambda$  do
9:   if  $|C_i| \leq \min\left(z + 1, \frac{|P_i| - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}\right)$  then
10:    OFFLINECLUSTER( $P_i, i, r$ )
11:    break
12:   end if
13: end for

```

Procedure 4 DELETE(p, r)

```
1:  $\lambda \leftarrow |\mathcal{F}_r|$ 
2: if  $p \in \mathcal{Z}_r$  then
3:   remove  $p$  from  $\mathcal{Z}_r$ 
4: else
5:   Let  $C_i$  be the cluster containing  $p$ , where  $i \in [\lambda]$ 
6:    $P_i \leftarrow (\cup_{i \leq j \leq \lambda} C_j) \cup \mathcal{Z}_r$ 
7:   if  $\mathcal{B}(c_i, 2r)$  covers  $> \min\left(z + 1, \frac{|P_i| - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}\right)$  from  $P_i$  then
8:     remove  $p$  from  $C_i$ 
9:   else
10:    OFFLINECLUSTER( $P_i, i, r$ )
11:   end if
12: end if
```

Procedure 5 OFFLINECLUSTER(P_i, i, r)

```
1:  $t'_r \leftarrow -1$ 
2: while  $i \leq k$  and  $P_i \neq \emptyset$  do
3:    $n_i \leftarrow |P_i|$ 
4:   Pick a uniform sample  $S_i$  of  $\psi \epsilon^{-1} k^2 \log k$  points from  $P_i$ 
5:   if  $z + 1 \leq \frac{n_i - z}{4(k - i + 1)}$  then
6:     Find a point  $p^* \in S_i$  such that  $|\mathcal{B}_{P_i}(p^*, 2r)| \geq \frac{n_i - z}{2(k - i + 1)}$ 
7:     if there does not exist such a  $p^*$  then
8:        $t'_r \leftarrow t + \frac{n_i - z}{2(k - i + 2)}$  // See Lemma E.1
9:     break
10:    end if
11:   else if  $z + 1 > \frac{n_i - z}{4(k - i + 1)}$  then
12:     Find a point  $p^* \in S_i$  such that  $|\mathcal{B}_{P_i}(p^*, 2r)| \geq \frac{n_i - z}{k - i + 1} - \frac{\epsilon z}{\beta k}$ 
13:     if there does not exist such a  $p^*$  then
14:        $t'_r \leftarrow t + \frac{\epsilon z}{2\beta k}$  // See Lemma E.2
15:     break
16:    end if
17:   end if
18:    $c_i \leftarrow p^*, C_i \leftarrow \mathcal{B}(p^*, 4r)$ , and  $P_{i+1} \leftarrow P_i \setminus C_i$ 
19:    $i \leftarrow i + 1$ 
20: end while
21:  $\mathcal{F}_r \leftarrow \{c_1, c_2, \dots, c_{i-1}\}$ ,  $\mathcal{L}_r \leftarrow \{C_1, C_2, \dots, C_{i-1}\}$ ,  $\mathcal{Z}_r \leftarrow P_i$ 
```

B Missing proofs Section 3.1

Lemma 3.4 (Procedure 3 maintains invariants). *Assume that at time t , we have point set P^t , data structure $\mathcal{D}_r = (\mathcal{F}_r, \mathcal{L}_r, \mathcal{Z}_r)$. We assume that the level and dense invariants hold at time t and $r \geq r_{\text{OPT}}^{t+1}$. At the start of time $t + 1$, we insert point p using Procedure 3. After the insertion, the level and dense invariants still hold with probability 1 if Procedure 5 was not called and with the probability of at least $1 - \frac{2(k-i+1)}{e^{\Psi \log k}}$, where $\Psi \geq 1$ if Procedure 5 was not called.*

Proof. If we enter the case in line 2, the new point p will be added to an existing cluster C_i . Then, we have that $p \in P_j$ for all $j \leq i$ and $p \notin P_j$ for all $j > i$. If we enter the case in line 4, the new point p is added as an outlier. This is the highest level, so we have that $p \in P_i$ for all i . The level invariant is maintained by definition in these cases. If we enter the case in line 9 and recluster levels i, \dots, k , we recluster the points in P_i as defined in line 7, and hence, none of the levels $j < i$ are affected. Then by Lemma 3.6, the level invariant is maintained for all levels. The dense invariant is maintained because of the check we do in line 9 for all levels. Furthermore, we choose the lowest level i where the dense invariant does not hold and recluster from this level upwards. Hence, the

dense invariant will hold for all levels $j < i$. We call Procedure 5 on P_i , so by Lemma 3.6, the dense invariant is maintained for levels $j \geq i$ with probability $1 - \frac{2(k-i+1)}{e^{\Psi \log k}}$, with $\Psi \geq 1$. \square

Lemma 3.5 (Procedure 4 maintains invariants). *Assume that at time t , we have point set P^t , instance $\mathcal{D}_r = (\mathcal{F}_r, \mathcal{L}_r, \mathcal{Z}_r)$, parameters $k, z \in \mathbb{N}$ and $\epsilon > 0$. We assume that the level and dense invariants hold at time t and $r \geq r_{\text{OPT}}^{t+1}$. At the start of time $t + 1$, we delete an arbitrary point p using Procedure 4. After the deletion, the level and dense invariants hold with probability 1 if Procedure 5 was not called, and with probability $1 - \frac{2(k-i+1)}{e^{\Psi \log k}}$ with $\Psi \geq 1$ if Procedure 5 was called.*

Proof. If we enter the case in line 2, where p is an outlier, it follows easily that the level and dense invariants are maintained. In the second case, starting in line 4, p is in some cluster C_i , either as a center or as another point. If C_i still covers sufficiently many points after the deletion of p as described in line 7, it follows easily that the level invariant is maintained. For the dense invariant, note that for all levels $j < i$, n_j and with this $\frac{n_j - z}{k - j + 1} - \frac{\epsilon z}{\alpha k}$ can only decrease. Furthermore, for all levels $j > i$, n_j remains unchanged. This observation, combined with the fact that for any $i \neq j$, $|C_j|$ is unchanged, means that the dense invariant still holds for all levels.

If C_i is no longer dense after deleting p , and we enter the case in line 9, the clusters in levels $j < i$ remain unchanged. The level invariant will be maintained since P_j is updated to $P_j \setminus \{p\}$ for all $j < i$. The dense invariant is maintained in levels $j < i$ by the same reasoning as the previous case. On the remaining points, Procedure 5 is called in line 10. Using Lemma 3.6, the level and dense invariants are maintained for the remaining levels with probability at least $1 - \frac{2(k-i+1)}{e^{\Psi \log k}}$, where $\Psi \geq 1$. Crucial for maintaining the level invariant is line 6. This line ensures that we do not consider the points in levels $j < i$ when constructing level i and any higher levels. \square

Lemma 3.6 (Procedure 5 maintains invariants with high probability). *Suppose the level and dense invariants hold for all levels $j < i$ and we call Procedure 5 on P_i as the result of an insertion or deletion. Let $\lambda \leq k$ be a random variable representing the number of levels we have after completing Procedure 5. If $r \geq r_{\text{OPT}}$, Procedure 5 maintains the level and dense invariants for all levels j with $i \leq j \leq \lambda$ with probability at least $1 - \frac{2(k-i+1)}{e^{\Psi \log k}}$, with $\Psi \geq 1$.*

Proof. By the while-loop structure in combination with line 18, the level invariant is maintained for all levels $j \geq i$. For each newly constructed level $j \geq i$, there are two cases within the while-loop.

For the first case where $z + 1 \leq \frac{n_j - z}{4(k-j+1)}$, we find a ball $\mathcal{B}_{P_j}(p^*, 2r)$ that covers $\geq \frac{n_j - z}{2(k-j+1)}$ points with probability at least $1 - \frac{2}{e^{\Psi \log k}}$ if $r \geq r_{\text{OPT}}$. For the proof of this, we refer to Lemma E.1.

Then, the dense invariant holds for any such level since

$$\frac{n_j - z}{2(k-j+1)} \geq \frac{n_j - z}{4(k-j+1)} \geq z + 1 \geq \min \left(z + 1, \frac{n_j - z}{k - j + 1} - \frac{\epsilon z}{\alpha k} \right). \quad (1)$$

In the second case where $z + 1 > \frac{n_j - z}{4(k-j+1)}$, we find a ball $\mathcal{B}_{P_j}(p^*, 2r)$ that covers $\geq \frac{n_j - z}{k - j + 1} - \frac{\epsilon z}{\beta k}$ points with probability $1 - \frac{2}{e^{\Psi \log k}}$ if $r \geq r_{\text{OPT}}$. For the proof of this, we refer to Lemma E.2. Since $\beta > \alpha$ (see Section 2.4), the dense invariant also holds for every level in this case. Then, the probability that the dense invariant holds for all levels j with $i \leq j \leq \lambda$ is at least $1 - \frac{2(k-i+1)}{e^{\Psi \log k}}$. This is because $\lambda - i + 1 \leq k - i + 1$ is the number of new levels we constructed during Procedure 5. \square

C Missing proofs Section 3.2

Lemma 3.8. *Let i be an iteration of the charging argument above such that we are in case 2. This means that $\mathcal{B}(c_1, 2r) \cup \dots \cup \mathcal{B}(c_i, 2r)$ does not intersect any of the remaining modified balls. Then, there must be a remaining modified ball covering $\leq \frac{n_i - z}{k - i + 1}$ points.*

Proof. We prove this using strong induction on iterations in case 2. First, let us consider the base case. In the base case, we are in iteration i which is the first iteration to be in case 2 of the charging argument. There have been $i - 1$ iterations before i which were charged according to case 1. For any

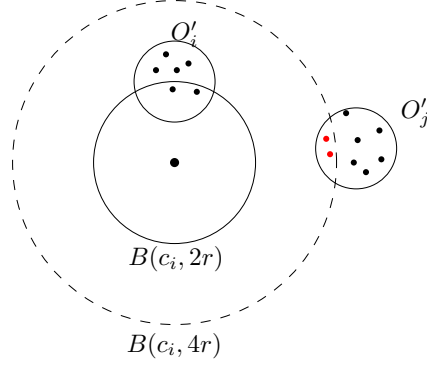


Figure 3: An example to illustrate the extra charging we do at the end of both cases 1 and 2. We are in iteration i , and the ball $B(c_i, 4r)$ covers two points from ball O'_j , with $j > i$. The two points, shown in red, will be charged to themselves as in charging *rule 1* and marked as covered.

$j < i$, define x_j to be the number of artificial outliers covered by $B(c_j, 4r)$. Then, for the remaining modified balls we have that:

$$|O'_i \cup \dots \cup O'_k| = n_i - \left(z - \sum_{j=1}^{i-1} (z_j^c - x_j) \right) - \sum_{j=1}^{i-1} (z_j^c - x_j) = n_i - z \quad (2)$$

This is because we assume that there are exactly z points outside our optimal solution $O_1 \cup \dots \cup O_k$. These z points are also outside $O'_1 \cup \dots \cup O'_k$ since for $1 \leq i \leq k$, $O'_i \subseteq O_i$. Of those z points, $(z - \sum_{j=1}^{i-1} (z_j^c - x_j))$ have not yet been seen in an iteration. Hence, these points are definitely not covered by $O'_i \cup \dots \cup O'_k$. Furthermore, we know that $\sum_{j=1}^{i-1} (z_j^c - x_j)$ artificial outliers have been defined in $O'_i \cup \dots \cup O'_k$ in total. This is because all iterations $j < i$ were in case 1, and any artificial outliers present in O'_j will be propagated to $(O'_{j+1} \cup \dots \cup O'_k) \cap P_{j+1}$, as these are included in z_j^c . We subtract x_j because these artificial outliers have already been counted by another z_j^c , and should not be counted again when computing the total amount of artificial outliers. Note that $z_j^d = 0$ for all iterations $j < i$ as there were no previous iterations before j in case 2. Since at most $n_i - z$ points are covered by $O'_i \cup \dots \cup O'_k$, there must be at least one of the modified optimal ball which covers $\leq \frac{n_i - z}{k - i + 1}$. This concludes the proof of the base case.

Now, consider the inductive step. We are in an iteration i that is in case 2. There have been $i - 1$ previous iterations, of which an arbitrary number has been charged by case 2. Assume that for any such iteration $j < i$ that was in case 2, we found an optimal ball O'_j that was covering at most $\frac{n_j - z}{k - j + 1}$ points. At the time of iteration j , the points of ball O'_j were charged to distinct points in $B(c_j, 2r)$, but not necessarily covered. In the iterations between j and i , however, more points of the ball O'_j may have been covered. Define $O'_{j,\text{unc}}$ as the points of O'_j that are still uncovered at the time of iteration i . Now, for the remaining modified balls, we should have that:

$$\begin{aligned} |O'_i \cup \dots \cup O'_k| &= n_i - \left(z - \sum_{j=1}^{i-1} (z_j^c - x_j) \right) - \sum_{j < i \text{ in case 1}} (z_j^c + z_j^d - x_j) \\ &\quad - \sum_{j < i \text{ in case 2}} (z_j^c + z_j^d - |O'_j| - x_j) - \sum_{j < i \text{ in case 2}} |O'_{j,\text{unc}}| \end{aligned} \quad (3)$$

Similar to the base case, $(z - \sum_{j=1}^{i-1} (z_j^c - x_j))$ points will definitely be outside $O'_i \cup \dots \cup O'_k$. In the iterations before $j < i$ in case 1, $(z_j^c + z_j^d)$ points are stored as a credit. We subtract x_j from this since we don't want to recount outliers from previous iterations. In each iteration $j < i$ that was in case 2, $(z_j^c + z_j^d - |O'_j|)$ are stored as a credit. Even though $B(c_j, 2r)$ is disjoint from the remaining modified balls in this case, $B(c_j, 2r)$ can still cover some artificial outliers present in the remaining modified balls. We do not want to recount these for the total amount of outliers in $O'_i \cup \dots \cup O'_k$, and hence we subtract x_j . Lastly, since the modified balls are disjoint, any points of O'_j with $j < i$ and iteration j in case 2 that are still in the point set n_i are not covered by $O'_i \cup \dots \cup O'_k$. We can rewrite

this as follows:

$$|O'_i \cup \dots \cup O'_k| = n_i - z - \left(\sum_{j < i} z_j^d + \sum_{j < i \text{ in case 2}} |O'_{j,\text{unc}}| \right) + \sum_{j < i \text{ in case 2}} |O'_j| = n_i - z \quad (4)$$

This is because by the definition of z_j^d , $(\sum_{j < i} z_j^d + \sum_{j < i \text{ in case 2}} |O'_{j,\text{unc}}|) = \sum_{j < i \text{ in case 2}} |O'_j|$. Since at most $n_i - z$ points are covered by $O'_i \cup \dots \cup O'_k$, there must be at least one of the remaining modified balls which covers $\leq \frac{n_i - z}{k - i + 1}$. This concludes the proof. \square

Lemma 3.9 (Coverage of $B(c_i, 2r)$). *When we are in case 2 of the charging argument for some iteration i , we must have that $|\mathcal{B}(c_i, 2r)| \geq \frac{n_i - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}$.*

Proof. We prove the statement by induction on iterations $i \leq k$ in case 2 of the charging argument. For the base case, iteration i is the first iteration in case 2. This means that for all previous iterations $j < i$, O'_j has been charged and covered by charging rule I. Hence, $\mathcal{B}(c_i, 2r)$ cannot intersect any such O'_j since the points in O'_j are not in P_i . We know that $\mathcal{B}(c_i, 2r)$ also does not intersect any of the remaining modified balls by our assumption that iteration i is in case 2. Hence, $\mathcal{B}(c_i, 2r)$ does not intersect $O'_1 \cup \dots \cup O'_k$. The ball $\mathcal{B}(c_i, 2r)$ can cover $x \leq \sum_{j=1}^{i-1} (z_j^c - x_j)$ artificial outliers, where x is the total number of artificial outliers present in $O'_i \cup \dots \cup O'_k$. The ball $\mathcal{B}(c_i, 2r)$ can cover at most $z - \sum_{j=1}^{i-1} (z_j^c - x_j)$ other points, since there are exactly z points outside $O_1 \cup \dots \cup O_k$, of which $\sum_{j=1}^{i-1} (z_j^c - x_j)$ have been seen in a previous ball $\mathcal{B}(c_j, 4r)$, for $j < i$. Hence, $|\mathcal{B}(c_i, 2r)| \leq z$. So, in order to satisfy the dense invariant, $|\mathcal{B}(c_i, 2r)| \geq \frac{n_i - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}$.

Now consider the induction case. We are in an arbitrary iteration i in case 2, and there have been an arbitrary number of iterations $j < i$ in case 2. For iterations $j < i$ in case 2, we assume that $|\mathcal{B}(c_j, 2r)| \geq \frac{n_j - z}{k - j + 1} - \frac{\epsilon z}{\alpha k}$. Even though $\mathcal{B}(c_i, 2r)$ is disjoint from the remaining modified balls, it can cover x artificial outliers and $z - \sum_{j=1}^{i-1} (z_j^c - x_j)$ other points. For x we have that:

$$x \leq \sum_{j < i \text{ in case 1}} (z_j^c + z_j^d - x_j) + \sum_{j < i \text{ in case 2}} (z_j^c + z_j^d - |O'_j| - x_j) \quad (5)$$

$\sum_{j < i \text{ in case 1}} (z_j^c + z_j^d - x_j)$ are the total number of artificial outliers from iterations in case 1, and $\sum_{j < i \text{ in case 2}} (z_j^c + z_j^d - |O'_j| - x_j)$ are the total number of artificial outliers from case 2. Then, for the total amount of points in $\mathcal{B}(c_i, 2r)$, we have:

$$|\mathcal{B}(c_i, 2r)| \leq z + \sum_{j < i} z_j^d - \sum_{j < i \text{ in case 2}} |O'_j| \leq z \quad (6)$$

This is because by the definition of z_j^d , we have that $\sum_{j < i} z_j^d \leq \sum_{j < i \text{ in case 2}} |O'_j|$. So, in order to satisfy the dense invariant, $|\mathcal{B}(c_i, 2r)| \geq \frac{n_i - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}$. This concludes the proof. \square

D Missing proofs Section 3.3

Lemma 3.10 (Duration of dense cluster for z is small). *Assume that we are currently at time t . Let us consider a level i in which $z + 1 \leq \frac{n_i^t - z}{4(k - i + 1)}$. Let $p = \arg \max_{p' \in S_i} |\mathcal{B}_{P_i}(p', 2r)|$, and $\mathcal{B}_{\max} = \mathcal{B}_{P_i}(p, 2r)$. Assume that $\frac{n_i^t - z}{2(k - i + 1)} \leq |\mathcal{B}_{\max}|$. Then, we can add $\mathcal{B}_{P_i}(p, 4r)$ as a cluster in our solution, and this cluster will be dense until time $t' = t + t^*$, with $t^* \geq \frac{n_i^t - z}{4(k - i + 1)}$.*

Proof. The subscript of n_i will be omitted for simplicity. At time t , we know that $\frac{n^t - z}{2(k - i + 1)} \leq |\mathcal{B}_{\max}|$. The resulting cluster $\mathcal{B}_{P_i}(p, 4r)$ will be dense as long as \mathcal{B}_{\max} covers more than $\min\left(z + 1, \frac{n^{t'} - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}\right)$ points at time t' . Hence, the dense invariant will not be broken as long as

\mathcal{B}_{\max} is greater than $z + 1$. Let t' be the time at which $|\mathcal{B}_{\max}| < z + 1$. We can lower bound the time t^* between t and t' using the change in size of \mathcal{B}_{\max} . Then, we get the following lower bound:

$$t^* \geq \frac{n^t - z}{2(k - i + 1)} - z \quad (7)$$

Then, since $z + 1 \leq \frac{n^t - z}{4(k - i + 1)}$ by our assumption, we have:

$$t^* > \frac{n^t - z}{2(k - i + 1)} - \frac{n^t - z}{4(k - i + 1)} \geq \frac{n^t - z}{4(k - i + 1)} \quad (8)$$

This concludes the proof. \square

Lemma 3.11 (Duration of dense cluster for z is large). *Assume that we are currently at time t . Let $z + 1 > \frac{n_i^t - z}{4(k - i + 1)}$ for some level i . Let $p = \arg \max_{p' \in S_i} |\mathcal{B}_{P_i}(p', 2r)|$, and $\mathcal{B}_{\max} = \mathcal{B}_{P_i}(p, 2r)$. Assume that $\frac{n_i^t - z}{k - i + 1} - \frac{\epsilon z}{\beta k} \leq |\mathcal{B}_{\max}|$. Then, we can add $\mathcal{B}_{P_i}(p, 4r)$ as a cluster in our solution, and this cluster will be dense until time $t' = t + t^*$, with $t^* = \Omega(\frac{\epsilon z}{k})$.*

Proof. The subscript of n_i will be omitted for simplicity. At time t , we know that $\frac{n^t - z}{k - i + 1} - \frac{\epsilon z}{\beta k} \leq |\mathcal{B}_{\max}|$. The resulting cluster $\mathcal{B}_{P_i}(p, 4r)$ will be dense as long as \mathcal{B}_{\max} covers more than $\min\left(z + 1, \frac{n^{t'} - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}\right)$ points at time t' , with α a fixed constant smaller than β . Hence, the dense invariant will not be broken as long as \mathcal{B}_{\max} covers at least $\frac{n^{t'} - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}$ points. Let t' be the time at which $|\mathcal{B}_{\max}| < \frac{n^{t'} - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}$. We can lower bound t^* by examining the change in size of \mathcal{B}_{\max} . Using that $n^{t'} \leq n^t + t^*$, we have:

$$\begin{aligned} t^* &\geq \left(\frac{n^t - z}{k - i + 1} - \frac{\epsilon z}{\beta k}\right) - \left(\frac{n^{t'} - z}{k - i + 1} - \frac{\epsilon z}{\alpha k}\right) \\ &\geq \frac{n^t - z}{k - i + 1} - \frac{n^t + t^* - z}{k - i + 1} - \frac{\epsilon z}{\beta k} + \frac{\epsilon z}{\alpha k} = \frac{-t^*}{k - i + 1} - \frac{\epsilon z}{\beta k} + \frac{\epsilon z}{\alpha k} \end{aligned} \quad (9)$$

Now, solving for t^* gives:

$$t^* \geq \frac{\epsilon z(k - i + 1)}{k(k - i + 2)} \left(\frac{1}{\alpha} - \frac{1}{\beta}\right) \geq \frac{\epsilon z}{2k} \left(\frac{1}{\alpha} - \frac{1}{\beta}\right) \quad (10)$$

Since α and β are two fixed constants such that $\beta > \alpha$, this gives $t^* = \Omega(\frac{\epsilon z}{k})$. This concludes the proof. \square

E Missing proofs Section 3.4

Lemma E.1 (Radius guess is small for z is small). *Assume that we are currently at time t . Let us consider a level i for which $z + 1 \leq \frac{n_i^t - z}{4(k - i + 1)}$. Let $p = \arg \max_{p' \in S_i} |\mathcal{B}_{P_i}(p', 2r)|$, where S_i is the sample chosen in Algorithm 5, and $\mathcal{B}_{\max} = \mathcal{B}_{P_i}(p, 2r)$. Assume that $|\mathcal{B}_{\max}| < \frac{n_i^t - z}{2(k - i + 1)}$. Then, with probability at least $1 - \frac{2}{e^\Psi \log k}$, for $\Psi \geq 1$, we have that $r < r_{OPT}$ and until time $t' = t + t^*$ we do not need to consider the instance for r , with $t^* \geq \frac{n_i^t - z}{2(k - i + 2)}$.*

Proof. Let O_1, \dots, O_k be the optimal balls at time t . Consider the same charging argument used in Lemma 3.7, where we charged one of the optimal balls to $\mathcal{B}_{P_i}(c_i, 4r)$ in each level i . Using Equation 4 and the fact that $O_i' \subseteq O_i$ for each of the remaining optimal balls, we know that $|O_i \cup \dots \cup O_k| \geq n_i^t - z$. Then, there must be at least one of the remaining optimal balls covering $\geq \frac{n_i^t - z}{k - i + 1}$ points. Let O_i be the largest such ball. Using the Chernoff bound, we show that with high probability, one of the sampled points $p \in S_i$ is in O_i . Define independent random variables $X_1, \dots, X_{|O_i|}$, one for each point in O_i . Each X_j , corresponding to point $j \in O_i$, will be 1 if $j \in S_i$

and 0 otherwise. Define $X = \sum_{j=1}^{|O_i|} X_j$. We know that $\mathbf{E}[X_j] = \frac{|S_i|}{n_i^t}$. Then, using linearity of expectation, we find:

$$\mathbf{E}[X] = \frac{|S_i| \cdot |O_i|}{n_i^t} \geq \frac{|S_i|}{n_i^t} \cdot \frac{n_i^t - z}{k - i + 1} \geq \frac{|S_i| \cdot (n_i^t - z)}{n_i^t \cdot k} \geq \frac{|S_i|}{k} \left(1 - \frac{z}{n_i^t}\right) \quad (11)$$

The condition $z + 1 \leq \frac{n_i^t - z}{4(k-i+1)}$ implies $z \leq \frac{n_i^t}{4(k-i+1)} \leq \frac{n_i^t}{4}$. Then, since $|S_i| = \psi \epsilon^{-1} k^2 \log k$, it follows that $\frac{|S_i|}{k} \left(1 - \frac{z}{n_i^t}\right) \geq \frac{3}{4} \psi \epsilon^{-1} k \log k$. Given that $\psi \geq 6\beta$ and $\beta > \alpha \geq 1$, this is at least $3\epsilon^{-1} k \log k \geq 3\Psi \log k$ for $\Psi \geq 1$.

Now, using the Chernoff bound, we find:

$$\Pr[|X - \mathbf{E}[X]| \geq \mathbf{E}[X]] \leq 2e^{-\frac{\mathbf{E}[X]}{3}} \leq \frac{1}{e^{\Psi \log k}}, \quad (12)$$

where $\Psi \geq 1$.

Hence, with probability of at least $1 - \frac{2}{e^{\Psi \log k}}$, there will be at least 1 point from O_i in S_i . Let us call this point p . If $r \geq r_{\text{OPT}}$, $\mathcal{B}_{P_i}(p, 2r)$ would cover all points of O_i since $p \in O_i$ and O_i has radius $\leq r_{\text{OPT}}$. However, since $|\mathcal{B}_{\max}| < \frac{n_i^t - z}{2(k-i+1)}$, we must have that $r < r_{\text{OPT}}$.

Left to prove is that until time $t' = t + t^*$ we do not need this instance of r , where $t^* \geq \frac{n_i^t - z}{2(k-i+2)}$. To this end, let \mathcal{B}_{\max}^t be \mathcal{B}_{\max} at time t . Consider the situation at time t . We know that $|\mathcal{B}_{\max}^t| < \frac{n_i^t - z}{2(k-i+1)}$.

Let t' be some time after t such that $|\mathcal{B}_{\max}^{t'}| \geq \frac{n_i^{t'} - z}{k-i+1}$. Hence, the instance for r becomes valid at time t' . We want to derive a lower bound for $t^* = t' - t$ to complete the proof. By examining the change in the size of \mathcal{B}_{\max} , we derive the following lower bound:

$$t^* \geq \frac{n_i^{t'} - z}{k - i + 1} - \frac{n_i^t - z}{2(k - i + 1)} \quad (13)$$

Using $n_i^{t'} \geq n_i^t - t^*$ and subsequently solving for t^* we find that:

$$t^* \geq \frac{n_i^t - z}{2(k - i + 2)} \quad (14)$$

This completes the proof. \square

Lemma E.2 (Radius guess is small for z is large). *Assume that we are currently at time t . Let us consider a level i in which $z + 1 > \frac{n_i^t - z}{4(k-i+1)}$. Let $p = \arg \max_{p' \in S_i} |\mathcal{B}_{P_i}(p', 2r)|$, where S_i is the sample chosen in Algorithm 5, and $\mathcal{B}_{\max} = \mathcal{B}_{P_i}(p, 2r)$. Assume that $|\mathcal{B}_{\max}| < \frac{n_i^t - z}{k-i+1} - \frac{\epsilon z}{\beta k}$. Then, with probability at least $1 - \frac{2}{e^{\Psi \log k}}$, for $\Psi \geq 1$, we have that $r < r_{\text{OPT}}$ and until time $t' = t + t^*$ we do not need to consider the instance for r , with $t^* \geq \frac{\epsilon z}{2\beta k}$.*

Proof. Let O_1, \dots, O_k be the optimal balls at time t . As in Lemma E.1, let O_i be the largest remaining optimal ball covering $\geq \frac{n_i^t - z}{k-i+1}$ points. Using the Chernoff bound, we show that with high probability, one of the sampled points $p \in S_i$ is in O_i . Define independent random variables $X_1, \dots, X_{|O_i|}$, one for each point in O_i . Each X_j , corresponding to point $j \in O_i$, will be 1 if $j \in S_i$ and 0 otherwise. Define $X = \sum_{j=1}^{|O_i|} X_j$. We know that $\mathbf{E}[X_j] = \frac{|S_i|}{n_i^t}$. Then, using linearity of expectation, we find

$$\mathbf{E}[X] = \frac{|S_i| \cdot |O_i|}{n_i^t} \geq \frac{|S_i|}{n_i^t} \cdot \frac{n_i^t - z}{k - i + 1}. \quad (15)$$

Without loss of generality, we can assume $\frac{n_i^t - z}{k-i+1} - \frac{\epsilon z}{\beta k} \geq 1$, otherwise there would not need to be any points in the ball $\mathcal{B}_{P_i}(p, 2r)$ to satisfy the dense invariant, and level i would be trivial. Using this, Equation (15) can be simplified as follows:

$$\mathbf{E}[X] \geq \frac{|S_i|}{n_i^t} \cdot \frac{n_i^t - z}{k - i + 1} \geq \frac{|S_i|}{n_i^t} \cdot \frac{\epsilon z}{\beta k} \quad (16)$$

We make a case distinction. Either $z \geq \frac{n_i^t}{2}$. In this case, Equation (15) further simplifies to

$$\mathbf{E}[X] \geq \frac{\epsilon |S_i|}{2\beta k}. \quad (17)$$

Recall that $|S_i| = \psi \epsilon^{-1} k^2 \log k$. Given that $\psi \geq 6\beta$ and $\beta > \alpha \geq 1$, we have

$$\mathbf{E}[X] \geq \frac{1}{2\beta} \psi k \log k \geq 3k \log k \geq 3\Psi \log k \quad (18)$$

for some $\Psi \geq 1$.

In the other case $z < \frac{n_i^t}{2}$, Equation (15) simplifies to

$$\mathbf{E}[X] \geq \frac{|S_i|}{n_i^t} \cdot \frac{n_i^t/2}{k-i+1} \geq \frac{|S_i|}{2k} \geq \frac{1}{2} \psi \epsilon^{-1} k \log k \geq 3\Psi \log k \quad (19)$$

for some $\Psi \geq 1$.

In both cases, using the Chernoff bound, we find that with probability at least $1 - \frac{2}{e^{\Psi \log k}}$, there will be at least one point p from O_i in X_i . If $r \geq r_{\text{OPT}}$, $\mathcal{B}_{P_i}(p, 2r)$ would cover all points of O_i since $p \in O_i$ and O_i has radius $\leq r_{\text{OPT}}$. However, since $|\mathcal{B}_{\max}| < \frac{n_i^t - z}{k-i+1} - \frac{\epsilon z}{\beta k} < \frac{n_i^t - z}{k-i+1} \leq |O_i|$, we must have that $r < r_{\text{OPT}}$.

Left to prove is that until time $t' = t + t^*$, with $t^* = \Omega(\frac{\epsilon z}{k})$, we do not need this instance of r . To this end, let \mathcal{B}_{\max}^t be \mathcal{B}_{\max} at time t . Consider the situation at time t . We know that $|\mathcal{B}_{\max}^t| < \frac{n_i^t - z}{k-i+1} - \frac{\epsilon z}{\beta k}$.

Let t' be some time after t such that $|\mathcal{B}_{\max}^{t'}| \geq \frac{n_i^{t'} - z}{k-i+1}$. The cluster $\mathcal{B}_{P_i}(p, 2r)$ in level i is now covering sufficiently many points. We want to derive a lower bound for $t^* = t' - t$ to complete the proof. By examining the change in the size of \mathcal{B}_{\max} and using that $n_i^{t'} \geq n_i^t - t^*$, we derive

$$t^* \geq \frac{n_i^{t'} - z}{k-i+1} - \left(\frac{n_i^t - z}{k-i+1} - \frac{\epsilon z}{\beta k} \right) \quad (20)$$

$$= \frac{n_i^t - t^* - z}{k-i+1} - \frac{n_i^t - z}{k-i+1} + \frac{\epsilon z}{\beta k} \quad (21)$$

$$= \frac{-t^*}{k-i+1} + \frac{\epsilon z}{\beta k} \quad (22)$$

as a lower bound for t^* . Solving for t^* gives

$$t^* \geq \frac{\epsilon z(k-i+1)}{\beta k(k-i+2)} \geq \frac{\epsilon z}{2\beta k}. \quad (23)$$

Since β is a constant, this is in $\Omega(\frac{\epsilon z}{k})$. This concludes the proof. \square

F Missing proofs from Section 3.5

Lemma F.1. *The amortized update time of our dynamic algorithm is $\mathcal{O}(\epsilon^{-3} k^6 \log(k) \log(\Delta))$.*

Proof. Let us fix an arbitrary time t and assume that at time t , there are $\lambda \leq k$ clusters. Let time $t' = t + t^*$ be the time at which we need to invoke Procedure 5 on an arbitrary level $i \leq k$, due to the dense invariant being violated. We have two cases:

- ① $z + 1 \leq \frac{n_i^t - z}{4(k-i+1)}$: For this case, Lemma 3.10 shows the following: If we find a cluster such that $|\mathcal{B}_{P_i}(c_i, 2r)| \geq \frac{n_i^t - z}{2(k-i+1)}$ at time t , then this cluster will remain dense for $t^* \geq \frac{n_i^t - z}{4(k-i+1)}$ update operations (either insert or delete).
- ② $z + 1 > \frac{n_i^t - z}{4(k-i+1)}$: For this case, Lemma 3.11 proves that if we find a cluster such that $|\mathcal{B}_{P_i}(c_i, 2r)| \geq \frac{n_i^t - z}{k-i+1} - \frac{\epsilon z}{\beta k}$ at time t , then this cluster will remain dense for the next $t^* \geq \frac{\epsilon z}{2k} (\frac{1}{\alpha} - \frac{1}{\beta})$ time steps. Asymptotically, $t^* = \Omega(\frac{\epsilon z}{k})$. By substituting the lower bound of z in this formula, we obtain $t^* = \Omega(\frac{\epsilon n_i^t}{k^2})$.

Observe that in both cases, we have a worst-case guarantee for the number of time steps during which the cluster $\mathcal{B}_{P_i}(c_i, 2r)$ remains dense. This, in turn, allows us to achieve a worst-case amortized update time, rather than the weaker notion of expected amortized update time. We analyze the update time of each case separately.

Let us start with the first case which is $z + 1 \leq \frac{n_i^t - z}{4(k-i+1)}$ and suppose that we find a cluster such that $|\mathcal{B}_{P_i}(c_i, 2r)| \geq \frac{n_i^t - z}{2(k-i+1)}$ at time t . We know that $t^* \geq \frac{n_i^t - z}{4(k-i+1)}$. The cost of reclustering levels i, \dots, k according to Lemma 3.3 is $\mathcal{O}(n_i^{t'} \epsilon^{-1} \cdot k^3 \log k)$, with $n_i^{t'} \leq n_i^t + t^*$. Then, the amortized update time of an arbitrary update operation is $\mathcal{O}(\frac{1}{t^*} (n_i^t + t^*) \epsilon^{-1} k^3 \log k) = \mathcal{O}(\frac{n_i^t}{t^*} \epsilon^{-1} k^3 \log k)$. Since $z \leq \frac{n_i^t}{4(k-i+1)}$, we obtain $\frac{n_i^t}{t^*} \leq n_i^t \cdot \frac{4(k-i+1)}{n_i^t - z} \leq 4k \cdot \frac{n_i^t}{n_i^t - z} \leq 4k \cdot \frac{n_i^t}{n_i^t (1 - \frac{1}{4(k-i+1)})} = 4k \cdot \frac{4(k-i+1)}{4(k-i+1)-1} = \mathcal{O}(k)$. Thus, $\mathcal{O}(\frac{n_i^t}{t^*} \epsilon^{-1} k^3 \log k) = \mathcal{O}(\epsilon^{-1} k^4 \log k)$.

Now, we consider the second case. Using a similar analysis as in the first case, we have $t^* = \Omega(\frac{\epsilon n_i^t}{k^2})$. Then, the amortized update time of an arbitrary update operation is $\mathcal{O}(\frac{n_i^t}{t^*} \epsilon^{-1} k^3 \log k) = \mathcal{O}(\epsilon^{-2} k^5 \log k)$.

After reclustering levels i, \dots, k , there is no longer a lower bound for t^* for levels $j < i$. Thus, it could happen that such a level j needs to be reclustered soon after time t' . Since $j < k$, this leads to an extra factor k such that the final amortized cost is $\mathcal{O}(\epsilon^{-2} k^6 \log k)$.

Finally, we need to consider the case that offline clustering fails for a level i . That is, we fail to find a ball covering sufficiently many points in Procedure 5. If offline clustering fails on some level i in case 1, meaning we fail to find a ball such that $|\mathcal{B}_{P_i}(c_i, 2r)| \geq \frac{n_i^t - z}{2(k-i+1)}$, then Lemma E.1 proves that with probability $1 - \frac{1}{e^{\Omega(\log k)}}$, the guess r is small (i.e., $r < r_{\text{OPT}}$) and indeed, remains small and we do not need to consider this guess until a time $t' = t + t^*$ where $t^* \geq \frac{n_i^t - z}{2(k-i+2)}$.

At time t' , we recluster levels i, \dots, k . Then, since we are in the first case, we have $z \leq \frac{n_i^{t'}}{4(k-i+1)}$. Thus, the amortized update time of an arbitrary update operation in this case is $\mathcal{O}(\frac{1}{t^*} (n_i^t + t^*) \epsilon^{-1} k^3 \log k) = \mathcal{O}(\frac{n_i^t}{t^*} \epsilon^{-1} k^3 \log k) = \mathcal{O}(\epsilon^{-1} k^4 \log k)$ using Lemma 3.3.

Next, we consider the situation when the offline clustering fails in case 2. Specifically, this occurs when we fail to find a ball such that $|\mathcal{B}_{P_i}(c_i, 2r)| \geq \frac{n_i^t - z}{k-i+1} - \frac{\epsilon z}{\beta k}$. According to Lemma E.2, the guess r is small (i.e., $r < r_{\text{OPT}}$) with probability $1 - \frac{1}{e^{\Omega(\log k)}}$ and indeed, remains small and we do not need to consider this guess until time $t' = t + t^*$ where $t^* = \Omega(\frac{\epsilon z}{k})$. This leads to an amortized update time of $\mathcal{O}(\epsilon^{-2} k^5 \log k)$. As mentioned above, the running time needs to be multiplied by an additional factor k to account for the possible need to recluster lower levels.

Finally, we need to consider $\mathcal{O}(\frac{\log \Delta}{\log(1+\epsilon)})$ guesses for the optimal radii. For small ϵ , this is in $\mathcal{O}(\frac{\log \Delta}{\epsilon})$. Thus, we derive the final amortized update time of $\mathcal{O}(\epsilon^{-3} k^6 \log k \log \Delta)$. \square

G How to support discrete k -center clustering with outliers

Lemma G.1. *Let (M, d) be a metric space and $\epsilon > 0$ be an accuracy parameter. The spread ratio $\Delta = \frac{d_{\max}}{d_{\min}}$ of all points ever inserted is assumed to be bounded. There exists a randomized fully dynamic algorithm that maintains a discrete k -center solution that allows up to $(1+\epsilon)z$ many outliers on the current set of points. At every point in time t , the current clustering with centers c_1, \dots, c_λ is a $(4+\epsilon)$ -approximation to an optimal solution for the (k, z) -center problem with high probability and $c_i \in P^t$ for all $i \leq \lambda$. Upon insertion or deletion of a point, the data structure is updated in amortized update time $\mathcal{O}(\epsilon^{-3} k^6 \log(k) \log(\Delta))$.*

Proof. Let r and i be fixed, and c_i be the center of cluster $C_i = \mathcal{B}_{P_i}(c_i, 4r)$. As long as c_i is not deleted, we report c_i as the i -th center. Note that any feasible solution for the discrete version is also a feasible solution for the non-discrete version. Therefore, the optimal radius for the discrete version is not smaller than the optimal radius for the non-discrete version.

After the deletion of the point c_i , we consider two cases: $\min\left(z + 1, \frac{n_i - z}{k - i + 1} - \frac{\varepsilon z}{\alpha k}\right) \leq 0$ or $\min\left(z + 1, \frac{n_i - z}{k - i + 1} - \frac{\varepsilon z}{\alpha k}\right) > 0$. If $\min\left(z + 1, \frac{n_i - z}{k - i + 1} - \frac{\varepsilon z}{\alpha k}\right) \leq 0$, then $n_i < (1 + \varepsilon)z$ as $i \geq 1$ and $\alpha > 1$. Hence, we can report all the n_i points as outliers and stop. We next consider the second case. The dense invariant states that $|\mathcal{B}_{P_i}(c_i, 2r)| \geq \min\left(z + 1, \frac{n_i - z}{k - i + 1} - \frac{\varepsilon z}{\alpha k}\right)$. Therefore, $|\mathcal{B}_{P_i}(c_i, 2r)| > 0$ holds in the second case, and there exists a point $\hat{p} \in \mathcal{B}_{P_i}(c_i, 2r)$. Then we report an arbitrary point $\hat{p} \in \mathcal{B}_{P_i}(c_i, 2r)$ after the deletion of c_i . Note that we do not replace c_i by \hat{p} in our data structure, and c_i does not change in our data structure as long as the i -th level is not reconstructed. The point \hat{p} is just the center that we report for the discrete version.

We next prove the 6-approximation guarantee. To do this, we show that $C_i \subseteq \mathcal{B}(\hat{p}, 6r)$. This implies that any feasible solution with radius $4r$ for the non-discrete version can be used to report a feasible solution for the discrete version. Also, see Figure 4 for a visual representation. Let $q \in C_i$. We show that $q \in \mathcal{B}(\hat{p}, 6r)$. Since $q \in C_i = \mathcal{B}(c_i, 4r)$, we have $d(q, c_i) \leq 4r$. Moreover, $d(c_i, \hat{p}) \leq 2r$ since $\hat{p} \in \mathcal{B}(c_i, 2r)$. Then by the triangle inequality, we have

$$d(q, \hat{p}) \leq d(q, c_i) + d(c_i, \hat{p}) \leq 4r + 2r = 6r.$$

To report a solution for the discrete version, it is enough to keep $\mathcal{B}(c_i, 2r) \cap P_i$. Note that our data structure already stores $C_i \cap P_i$, and since $\mathcal{B}(c_i, 2r) \cap P_i \subseteq C_i \cap P_i$, the space complexity and time complexity remain the same. \square

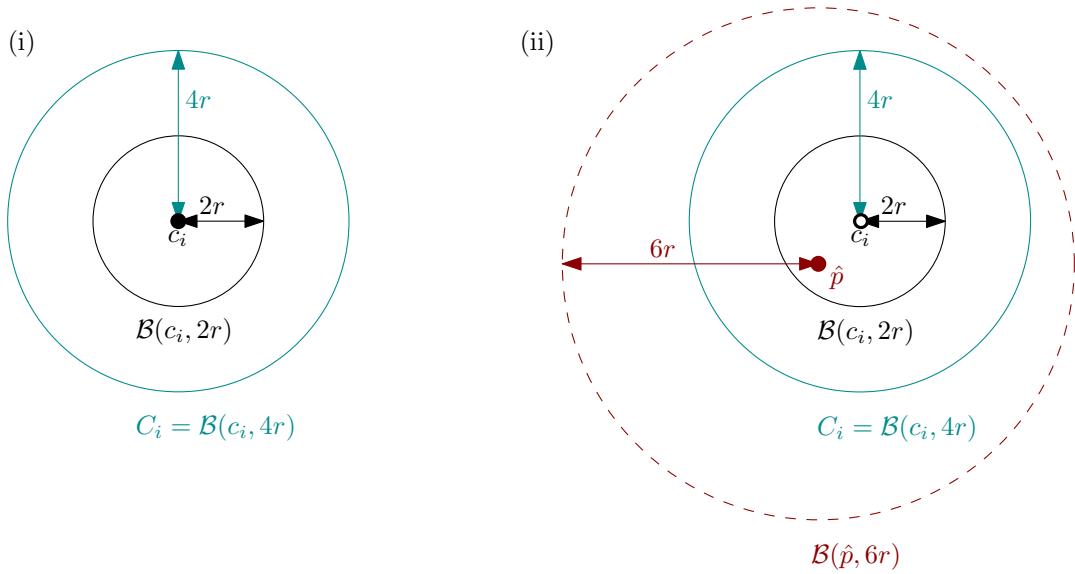


Figure 4: Illustration of the claim's proof. Part (i): before deletion of c_i , we report c_i as the center. Part (ii): after deletion of c_i , we report an arbitrary point $\hat{p} \in \mathcal{B}(c_i, 2r)$ as the center. Then, $\mathcal{B}(\hat{p}, 6r)$ can cover all the points in the cluster C_i .

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The problem setting stated at the beginning of the abstract summarizes this paper's scope as formalized in Section 1.3. The main statements made in the abstract can be found in Theorem 1.1 in Section 1.1, which is a summarization of Lemma 3.7 in Section 3.2 and Lemma F.1 in Section 3.5.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: As stated in the abstract, Section 1.1, Section 1.3 and the result Lemma 3.7, our algorithm yields a bicriteria approximation. This means that we approximate the objective function and also allow a slight violation of the outlier side constraint while comparing with an optimal solution that fulfills the side constraint exactly. Our results hold with high probability, which is stated formally in statement Lemma 3.6.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The problem setting alongside assumptions are stated in Section 1.3. This paper's main statements are Lemma 3.7 and Lemma F.1. Lemma 3.8 and Lemma 3.9 are used to prove Lemma 3.7. It uses the assumption that the level and dense invariants hold. Lemma 3.6, Lemma 3.4 and Lemma 3.5 show that running our procedures maintains these invariants. Lemma F.1 uses Lemma 3.3, Lemma 3.11, Lemma 3.10, Lemma E.1, Lemma E.2.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [NA]

Justification: The paper does not contain experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

- (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: The paper does not contain experiments.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [NA]

Justification: The paper does not contain experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: The paper does not contain experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [NA]

Justification: The paper does not contain experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: As our paper is of theoretical nature, most of the points addressed in the Code of Ethics do not apply. Further, we could not identify any foreseeable dangers or harms directly caused by the utilization of our results.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: There is no foreseeable direct societal impact of the work performed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: The paper does not use existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.