Coreset for Line-Sets Clustering

Sagi Lotan sagi.lotan@gmail.com
Ernesto Evgeniy Sanches Shayda ernestosanches@gmail.com
Dan Feldman dannyf.post@gmail.com

Robotics & Big Data Labs,
Computer Science Department, University of Haifa

Abstract

The input to the line-sets k-median problem is an integer $k \geq 1$, and a set $\mathcal{L} = \{L_1, \ldots, L_n\}$ that contains n sets of lines in $\mathbb{R}^d$. The goal is to compute a set C of k centers (points in $\mathbb{R}^d$) that minimizes the sum $\sum_{L \in \mathcal{L}} \min_{\ell \in L, c \in C} \operatorname{dist}(\ell, c)$ of Euclidean distances from each set to its closest center, where $\operatorname{dist}(\ell, c) := \min_{x \in \ell} \|x - c\|_2$. An ε*-coreset* for this problem is a weighted subset of sets in $\mathcal{L}$ that approximates this sum up to $1 \pm \varepsilon$ multiplicative factor, for every set C of k centers. We prove that *every* such input set $\mathcal{L}$ has a small ε-coreset, and provide the first coreset construction for this problem and its variants. The coreset consists of $O(\log^2 n)$ weighted line-sets from $\mathcal{L}$, and is constructed in $O(n \log n)$ time for every fixed $d, k \geq 1$ and $\varepsilon \in (0, 1)$. The main technique is based on a novel reduction to a "fair clustering" of colored points to colored centers. We then provide a coreset for this coloring problem, which may be of independent interest. Open source code and experiments are also provided.

1 Introduction

In the classic k*-mean* clustering problem, the input is a set P of n points in a metric space $(\mathcal{X}, \operatorname{dist})$, and an integer $k \geq 1$. The goal is to compute a set C^* of k centers (points in $\mathcal{X}$) that minimizes the sum of squared distances over each point $p \in P$ to its nearest center in C^*, i.e., to compute

$$C^* \in \underset{C \subseteq \mathcal{X}, |C| = k}{\arg\min} \sum_{p \in P} \tilde{D}(p, C),$$

where $\tilde{D}(p, C) := \min_{c \in C} \operatorname{dist}^2(p, c)$. This problem is arguably the most common clustering problem formulation, both in industry and academy; see references e.g. in [5, 20, 31, 36].

A natural generalization is to replace this input set P of n points by a set $\mathcal{P}$ of n sets in $\mathcal{X}$. The distance from such an input set $P \in \mathcal{P}$ to a set C^* of centers can then be defined as the distance between the closest point-center pair. This problem is called k*-mean for sets;* see e.g. [26] and references therein. Its goal is to compute

$$C^* \in \underset{C \subseteq \mathcal{X}, |C| = k}{\arg\min} \sum_{P \in \mathcal{P}} \tilde{D}(P, C), \tag{1}$$

where $\tilde{D}(P, C) := \min_{p \in P} \tilde{D}(p, C)$.
A special case is where every such set $P \in \mathcal{P}$ is a line ℓ in $\mathcal{X} = \mathbb{R}^d$. This problem is called k*-mean for lines*; see e.g. [10, 17, 35]. Its goal is to compute a set C^* of k points in $\mathbb{R}^d$ such that

$$C^* \in \underset{C \subseteq \mathbb{R}^d, |C| = k}{\arg\min} \sum_{\ell \in \mathcal{P}} \tilde{D}(\ell, C). \tag{2}$$

This paper considers a further generalization of the last problem, where the input is a set of n *sets*, each contains *multiple* lines as follows.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

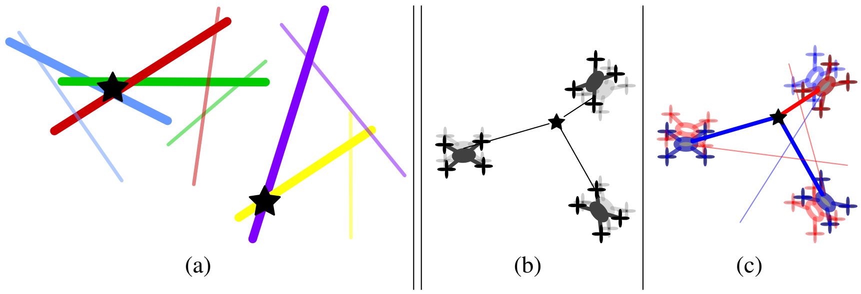


Figure 1: ***Line-sets Clustering:****(a) The $k = 2$ line-sets mean (black stars) of $n = 5$ sets (in different colors), each contains a pair ($m = 2$) of lines (same color) in the plane ($\mathcal{X} = \mathbb{R}^2$). The closest line to the stars in each pair of lines is bold.* ***Application:*** *(b) A set of $n = 3$ drones, each equipped with an on-board 2D camera, takes a snapshot of a point in $\mathbb{R}^3$ (the black $k = 1$ star). Due to the missing depth, each such point corresponds to a pixel in the 2D image's plane, or a line in $\mathbb{R}^3$ which is the intersection of the image's plane with the camera's pin-hole. (c) The estimated position of the drone from the noisy observations is inaccurate, so we use $m = 2$ guesses or consecutive images over time (blue and red line). One of them (in bold) is closer to the observed point.*

Line-sets clustering. The input to the line-sets k-mean problem is an integer $k \geq 1$, and a set $\mathcal{L} = \{L_1, \ldots, L_n\}$ of n sets, where each set $L = \{\ell_1, \ldots, \ell_m\}$ in $\mathcal{L}$ consists of $m \geq 1$ lines in $\mathbb{R}^d$. The goal is to compute a set C^* of k points in $\mathbb{R}^d$ that minimizes its sum of squared distances over every set of lines in $\mathcal{L}$, i.e.,

$$C^* \in \operatorname*{arg\,min}_{C \subseteq \mathbb{R}^d, |C|=k} \sum_{L \in \mathcal{L}} \tilde{D}(L, C). \tag{3}$$

Here, $\tilde{D}(L, C) := \min_{\ell \in L} \tilde{D}(\ell, C)$ is the shortest squared distance between a point in C to its closest line in L; see Fig 1.

1.1 Motivation

The classic k-clustering of points has numerous applications, and the k line-sets clustering is its natural generalization. It is thus not surprising that these applications can be easily generalized and extended. We give examples from different research fields as follows.

Handling missing data. A natural application is clustering of a data with missing values, as described e.g. in [10]. Here, each record in the input data-set is represented by a point in d dimensions. If an entry is missing, we may replace it by all possible values, resulting in a line in $\mathbb{R}^d$ that is parallel to one of the axes. Similarly, if the missing entry is discrete, i.e., can be one in a finite set of options, it results in a set of points. Our paper handles combination of them both. An example of this can be found in Dataset 4 in our experimental results for California housing. The entries may be the number of the apartment or floor, which corresponds to a set of values. A monthly rent is a continuous variable, which corresponds to a line. Together they correspond to line-sets. Another perspective would be of sets clustering with missing values, where most of the applications in [26] may be generalized using our work.

Computer vision. In the fundamental 3D model reconstruction problem, we are given n pixels (features). Each pixel is from a different 2D image that captures the same point in $\mathbb{R}^3$, i.e., the real world. Such a 2D pixel corresponds to a line in $\mathbb{R}^3$ that passes through the camera's pinhole and the pixel in the image plane. The goal is to estimate the location of the point in $\mathbb{R}^3$, based on the set of n 2D images. The task of estimating a 3D point location given a single set of lines is called *Triangulation*. Due to uncertainty and noise, feature extraction algorithms usually identify $m \geq 1$ pixels in each image as the captured point, without certainty about which of the pixels are noise and which are correct. Here, the center is this desired point in $\mathbb{R}^3$, and every image is represented by m lines. To obtain a model or a "3D point-cloud" [18, 22] we may wish to compute k points (centers) of the 3D object. See Fig. 1 (right).

Colored point-sets clustering applications. This paper reduces the line-sets clustering problem to clustering sets of colored points, which has its own applications. An example is the facility location problem, as in the home and work example in Section 3. Another application is text documents clustering, as explained in Section 5. Here, every document is represented by a set of colored vectors, where each vector corresponds to a paragraph in a "bag of words" representation. Each color represents a different type of paragraph, e.g., introduction, summary, main section, theorem, etc.

More generally, the problem is relevant where each facility is useful for a specific subset of input clients.

We hope that this paper will open the research toward more complex and general shapes that may yield better approximations to high-ways and polygons such as k segments (as in [38, 25, 10]), arcs or k-subspaces instead of lines.

1.2 Generalizations and computation models

Our main technique to handle the problems above, and especially their streaming, distributed, and parallel versions, is to design data summaries (often called coresets) for the k line-sets clustering problem, and its corresponding set of queries. We may then run naive algorithms such as exhaustive search on the small coreset to extract the approximated solution, or apply existing heuristics and obtain faster results.

In the following definition, a pseudo distance $\tilde{D}(p,q)$ assigns a non-negative number for any pair $p,q \in R^d$. The pseudo distance between sets is then the distance between the closest pair, as in (1) above. See exact definition and constraints in Definition 2.1.

Definition 1.1 (Coreset)**.** *Let $k \geq 1$ be an integer, and $\mathcal{L} = \{L_1, \ldots, L_n\}$ be an input set, where each $L = \{\ell_1, \ldots, \ell_m\} \in \mathcal{L}$ consists of $m \geq 1$ lines in $\mathbb{R}^d$. For a given approximation error parameter $\varepsilon \in (0,1)$ and an integer $k \geq 1$, an* ε-coreset of $\mathcal{L}$ for k line-sets clustering *over a pseudo distance $\tilde{D}$ is a weighted set $\mathcal{S} \subseteq \mathcal{L}$ that approximates the sum of pseudo distances to $\mathcal{L}$ from* every *set C of k centers in $\mathbb{R}^d$, up to a multiplicative factor of $1 \pm \varepsilon$. More precisely, there is a* weight function $v : \mathcal{S} \to [0, \infty)$ *such that for every set $C \subseteq \mathbb{R}^d$ of size $|C| \leq k$,*

$$\left| \sum_{L \in \mathcal{L}} \tilde{D}(L,C) - \sum_{L \in \mathcal{S}} v(L)\tilde{D}(L,C) \right| \leq \varepsilon \sum_{L \in \mathcal{L}} \tilde{D}(L,C).$$

In particular, an (optimal) line-sets k-mean C^* of the weighted set $(\mathcal{S}, v)$ is a $(1+\varepsilon)$-approximation to the k-line-sets mean of $\mathcal{L}$. This is the strongest type of coresets (sometimes called "strong coreset" [33, 41]), in the sense that it approximates *every* query C, and not just, say, the optimal solution. It is also a weighted subset of the input, unlike a sketch, linear combination, or general subsets of lines in $\mathbb{R}^d$ which are not a subset of $\mathcal{L}$; see next Section 1.3.

1.3 Related work

Clustering n points by k center points is a fundamental problem in machine learning [40]. Applications can be found in operations research [6, 8], statistics [20] and computational geometry [4, 28, 43], including constant factor approximations (randomized [2] and deterministic [7]). Many of the recent results [13, 21] are based on coresets (under different definitions) that can usually be computed in time that is near-linear in the input size and number k of centers. The size of these coresets is usually $(k/\varepsilon)^{O(1)}$, where $\varepsilon \in (0,1)$ is a parameter that represents the approximation error.

For the special case $m = 1$, our coreset in Definition 1.1 is for k-means of n lines. Such a coreset was suggested in [35] whose size is $\log^2(n)k^{o(k)} \cdot \varepsilon^{-2}$, for any constant $\varepsilon > 0$, and is improved in our paper. For the case that the lines are parallel to the axes, coresets were suggested in [10]. These results also hold for j-dimensional linear subspaces, but only if they are parallel to the axes of $\mathbb{R}^d$. The main motivation is handling j missing entries in each database's record (point).

Coresets for input sets of points, lines, and fair clustering appeared relatively recently. Coreset for (point)-sets clustering as in Eq. (1), where each set consists of m points, was suggested in [26] and its size is $\left(\frac{\log n}{\varepsilon}\right)^2 k^{O(m)}$.

Our main technical result is a coreset for colored (points) clustering, where the sets and centers are colored, and each input point can be served by a center of the same color. This approach is strongly related to fair clustering, where each group is represented by a color.

Variations of fair clustering were suggested in [12, 23, 27] and references therein. Coresets for fair clustering were suggested in [39]. This paper generalizes some of these results in the sense that the input points are colored sets, but the centers are also colored.

Our coreset constructions, as many of the recent coreset constructions, is based on the Feldman-Langberg framework [9, 11, 15, 29]. This framework reduces the problem of computing a coreset – to the problem of computing the importance (sensitivity) for each of the n input points/sets.

1.4 Main contributions

We suggest the first coreset construction for line-sets clustering whose size (number of weighted sets) is sub-linear in the number n of input sets. This is by a reduction to another problem, fair clustering (colored points) with colored centers, which may be of independent interest. We then suggest a coreset for this type of fair clustering problems. More precisely, we provide the following contributions for every pseudo distance $\tilde{D}$, constant approximation error $\varepsilon \in (0, 1)$, and constant integers $d, k, m \geq 1$. See the corresponding theorems for exact bounds.

(i) An ε-coreset $S \subseteq \mathcal{L}$ for line-sets clustering of any set $\mathcal{L}$ of n sets , each consists of at most m lines; see Theorem 4.1

(ii) An ε-coreset $S \subseteq \mathcal{P}$ for colored sets clustering of any set $\mathcal{P}$ of n sets in $\mathcal{X}$; See Theorem 4.2

Each of these coresets has size $|\mathcal{S}| \in O(\log^2 n)$ and can be computed in $O(n \log n)$ time, with high probability. The following results are straightforward implication of the above results, using existing merge-reduce frameworks [1]:

(iii) Support for streaming data in insertion time and memory that is poly-logarithmic in the number n of the sets seen so far in the stream.

(iv) Support for parallel computations on data that may be distributed on M machines using $1/M$ amortized insertion time.

(v) Support for deletion of sets from the stream in poly-logarithmic time during the streaming, but using memory that is linear with n.

(vi) FPTAS in time $O(n \log n)$, by running an exhaustive search on the corresponding small corsets.

Dependencies on input parameters. Our first result above is in fact an LTAS (linear time approximation scheme), since the asymptotic running time depends polynomially on $1/\varepsilon$, and near-linear in n. The dependency on d is polynomial, and the exponential dependency on k and m is unavoidable due to known lower bounds for special cases e.g. in [19, 26]. These are worst-case bounds, and the actual approximations errors are much smaller in practice as shown in the experimental results; see Section 5.

1.5 Novel technique: from line-sets to fair clustering

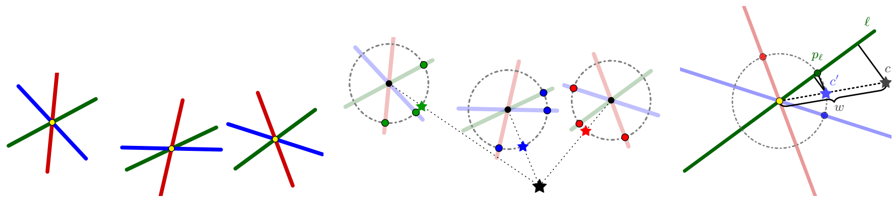
Existing coresets for point-sets clustering are heavily based on the triangle inequality between pair of points, which is not satisfied for the case of lines. On the other hand, existing coresets for lines were reduced to coresets for weighted centers, which do not support sets as an input. The main challenge in this paper is to combine these two results. This is by suggesting a novel reduction and coresets for colored-weighted centers, which may be of independent interest and borrows techniques from fair clustering. To our knowledge, this is the first link of this type between geometric shapes and fairness. We hope it will open the door for many other coresets; see Section 6.

Our reduction is based on the following three steps.

Grouping. Algorithm 3 in Section 3.2 recursively chooses a very "dense" constant fraction of sets $\mathcal{L}^m$ that are close to m center points (robust medians) $\mathcal{B}^m = \{b_1, \ldots, b_m\}$; see Definition 2.5. The ith line in each of these sets is then translated to its median b_i, for every $i \in [m]$. At this stage we reduced our problem to compute sensitivity for $|\mathcal{L}^m| \in O(n)$ sets of lines that form m "stars"; see Fig. 2(a).

Reduction to points. The distance from each line that intersects a robust median b_i (say, the origin) to a query center $c_i \in C$ is its distance to a unit vector in the same direction, weighted by the norm of the vector; see Fig. 2(b). This distance can be approximated, up to a constant factor, by replacing c_i with its projection on the sphere $c_i / \|c_i\|$ and its antipodal point $-c_i / \|c_i\|$.

Colored sets and centers. The new problem is now to compute sensitivity for n sets, each contains m *colored points* on m unit spheres; see Fig. 2(c). Each center in the set C of k queries is duplicated m times in m colors. The result is a set of $2mk$ weighted centers on m unit sphere with an additional constraint: the ith center can "serve" only the lines on the ith sphere.



(a) (b) (c)

Figure 2: **(a)** *A set $\mathcal{L}$ of 3 triplets of lines, each in a different color. The first line in each triplet intersects the same yellow (robust median) point b_1, and similarly for the other lines and medians b_2, b_3.* **(b)** *The distance from every colored line to a query center $c \in \mathbb{R}^2$ (black star) is the same as its distance from the star in the same color multiplied by the distance w from the center to its median.* **(c)** *The distance between a line ℓ (in green) and the point c (black) is the distance between c' (blue) and ℓ multiplied by w (black brackets). This distance is approximated by the distance $\|c' - p_\ell\|$ as stated in Claim C.3.*

2 Preliminaries

The results in this paper hold not only for squared distance functions but for other functions such as non-squared distances or M-estimators. To generalize this notion, we use the following pseudo distance over a metric space $(\mathcal{X}, \mathrm{dist})$; see many examples of such functions in [26]. From Section 3.2, i.e., for the case of input lines, we assume that $\mathcal{X} = \mathbb{R}^d$ and dist is the Euclidean norm. That is, the metric space is $(\mathbb{R}^d, ||.||_2)$ but lip below can still be any function that satisfies the following log-log Lipschitz condition.

Definition 2.1 (Pseudo-distance $\tilde{D}$)**.** *Let* $\mathrm{lip} : [0, \infty) \to [0, \infty)$ *be a* non-decreasing function *that satisfies the following condition: There is a constant $r \in (0, \infty)$ such that for every $x, z > 0$ we have $\mathrm{lip}(zx) \leq z^r \mathrm{lip}(x)$. Let $(\mathcal{X}, \mathrm{dist})$ be a metric space, and $\tilde{D}$ be a function that maps every pair of points $p, c \in \mathcal{X}$ to*

$$\tilde{D}(p, c) := \mathrm{lip}(\mathrm{dist}(p, c)).$$

For a pair of finite sets $P, C \subseteq \mathcal{X}$, denote $\tilde{D}(P, C) := \min_{p \in P, c \in C} \tilde{D}(p, c)$, $\tilde{D}(P, c) := \tilde{D}(P, \{c\})$ and $\tilde{D}(p, C) = \tilde{D}(\{p\}, C)$.

The motivation behind the last definition is that it satisfies the following pair of properties.

Lemma 2.2 (weak triangle [42])**.** *Let $\mathcal{X}$, $\tilde{D}$, and r be as defined in Definition 2.1. Then $\tilde{D}$ satisfies the following ("weak triangle") inequalities (i)–(ii) for every $p, q, c \in \mathcal{X}$:*

(i) For $\rho = \max\{2^{r-1}, 1\}$, $\quad \tilde{D}(p, q) \leq \rho(\tilde{D}(p, c) + \tilde{D}(c, q))$.

(ii) For $\phi = (4r - 4)^{r-1}$, $\quad \tilde{D}(p, c) - \tilde{D}(q, c) \leq \phi \tilde{D}(p, q) + \frac{\tilde{D}(p,c)}{4}$.

In the rest of the paper, for an integer $n \geq 1$ we denote $[n] = \{1, \ldots, n\}$. Also, unless otherwise stated $\tilde{D}, \mathcal{X}, r$ are as stated in Definition 2.1.

For both the line-sets clustering and the colored-sets clustering problems/coresets, the input is a set of n sets, each of size m. We call it (n, m)-set for short as follows.

Definition 2.3 ((n, m)-set)**.** *For a given integer $m \geq 1$, an* m-set *P is a set of m items, i.e. $|P| = m$. For an additional integer $n \geq 1$, an* (n, m)-set *is a set of n m-sets.*

Definition 2.4 ((n, m)-ordered-set)**.** *An m-ordered set P is an ordered-set of $m \geq 1$ points in $\mathcal{X}$. An (n, m)-ordered-set is a set of $n \geq 1$ m-ordered-sets.*

Informally, a robust median for an optimization problem at hand is an element b that approximates the optimal value of this optimization problem, with some leeway on the number of input elements considered. In the context of facility location, the facility (center) needs to serve only a subset of the closest clients (input points).

Let $\mathcal{P}$ be an (n, m)-set, $C \subseteq \mathcal{X}$ and $\gamma \in (0, 1]$. We denote by $\mathrm{closest}(\mathcal{P}, C, \gamma)$ the set that is the union of $\lceil \gamma |\mathcal{P}| \rceil$ sets $P \in \mathcal{P}$ with the smallest value of $\tilde{D}(P, C)$, i.e.,

$$\mathrm{closest}(\mathcal{P}, C, \gamma) \in \mathop{\arg\min}_{\mathcal{Q} \subseteq \mathcal{P} : |\mathcal{Q}| = \lceil \gamma |\mathcal{P}| \rceil} \sum_{P \in \mathcal{Q}} \tilde{D}(P, C). \tag{4}$$

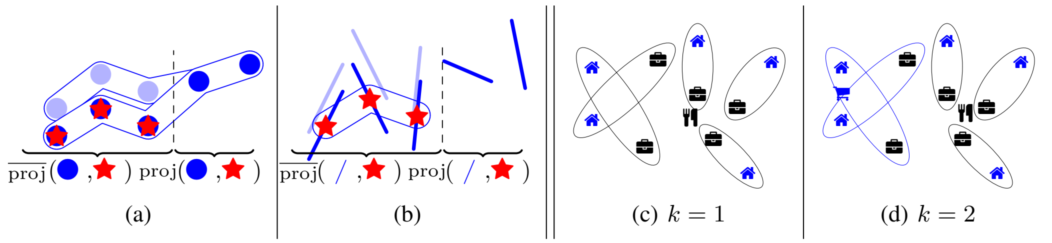


Figure 3: ***Example sets projections for Definition 2.7:*** **(a)** *The projection of $P =$ ● (top 5 points) onto the ordered set $\mathcal{B} =$ ★ (red stars) yields $T(P, \mathcal{B}) =$ ● (5 dark blue points), i.e., the union of 3 points $\text{proj}(P, \mathcal{B})$ (3 on the stars) with the 2 remaining points $\overline{\text{proj}}(P, \mathcal{B})$.* **(b)** *The projection of $L =$ / (top 5 lines) onto the ordered set $\mathcal{B}$ (red stars) yields $T(L, \mathcal{B})$ (5 dark blue lines), i.e., the union of 3 lines $\text{proj}(P, \mathcal{B})$ (translated to to the stars) with the 2 remaining lines $\overline{\text{proj}}(L, \mathcal{B})$.* **Colored-set k-clustering:** *The input is a set of $n = 5$ pairs of points that correspond to home/work addresses (in blue and black).* **(c)** *$k = 1$ center, which is a restaurant (black fork and knife). Its distance to a person is the distance to her office (black suitcase).* **(d)** *$k = 2$ centers: a restaurant (black fork and knife) and a grocery shop (blue cart). Each person chooses the closest restaurant to her office or the closest grocery to her home.*

For simplicity of notation, we define $\text{closest}(\mathcal{P}, C) := \text{closest}(\mathcal{P}, C, \frac{1}{|\mathcal{P}|})$ as one of the closest items to C in $\mathcal{P}$. Here and in the rest of the paper ties are broken arbitrarily.

Definition 2.5 (Robust median [16])**.** *Let $\mathcal{P}$ be an (n, m)-set in $\mathcal{X}$, $\gamma \in (0, 1]$, and*

$$\tilde{D}^*(\mathcal{P}, \gamma) = \min_{b \in \mathcal{X}} \sum_{P \in \text{closest}(\mathcal{P}, \{b\}, \gamma)} \tilde{D}(P, b).$$

*For $\tau \in (0, 1)$ and $\alpha \geq 0$, a point $b \in \mathcal{X}$ is a (γ, τ, α)-*median *for $\mathcal{P}$ if*

$$\sum_{P \in \text{closest}(\mathcal{P}, \{b\}, (1-\tau)\gamma)} \tilde{D}(P, b) \leq \alpha \cdot \tilde{D}^*(\mathcal{P}, \gamma).$$

Definition 2.6 (Translation)**.** *Let b be a point in $\mathcal{X}$. Than (i) For every $p \in \mathcal{X}$, we define $T(p, b) := b$, and (ii) If $\mathcal{X} = \mathbb{R}^d$, for every line ℓ in $\mathbb{R}^d$ we define $T(\ell, b)$ to be a line parallel to ℓ that intersects b.*

In what follows, we define the projection of a set (points or lines) over a set of points; see Fig. 3.

Definition 2.7 (Set projection)**.** *Let m, j be a pair of integers such that $1 \leq j \leq m$. Let P be either an m-set of points in $\mathcal{X}$ or an m-set of lines in $\mathcal{X} = \mathbb{R}^d$. Let $\mathcal{B} = (b_1, \ldots, b_j)$ be an ordered j-set of points in $\mathcal{X}$. Let $p_1 \in P$ denote the closest item $\{p_1\} = \text{closest}(P, \{b_1\})$ to b_1. For every integer $i \in \{2, \ldots, j\}$, recursively define p_i to be the closest item $\{p_i\} = \text{closest}(P \setminus \{p_1, \ldots, p_{i-1}\}, \{b_i\})$ in $P \setminus \{p_1, \ldots, p_{i-1}\}$ to b_i. We denote,*

(i) $\text{proj}(P, \mathcal{B}) := \{\mathrm{T}(p_1, b_1), \ldots, \mathrm{T}(p_j, b_j)\}$ *as the j items in P that were projected onto $\mathcal{B}$.*

(ii) $\overline{\text{proj}}(P, \mathcal{B}) := P \setminus \{p_1, \ldots, p_j\}$ *as the $m - j$ items in P that were not projected onto $\mathcal{B}$.*

(iii) $\mathrm{T}(P, \mathcal{B}) := \text{proj}(P, \mathcal{B}) \cup \overline{\text{proj}}(P, \mathcal{B})$ *as the* projection *of P onto $\mathcal{B}$.*

(iv) For projection over an empty set, we define $\overline{\text{proj}}(P, \emptyset) := \mathrm{T}(P, \emptyset) := P$.

3 Sensitivity

Our main coreset construction (Algorithm 4) is a standard non-uniform sampling algorithm based on Feldman-Langberg framework [14]. The main challenge in this framework is to compute the importance of each input set, known as sensitivity. Our main contribution is an algorithm that computes this sensitivity with provable guarantees on its running time and the total sensitivity, which implies the size of the resulting coreset.

As explained in the novelty section, we provide coreset construction for line-sets, by bounding sensitivity of a constant fraction of the input lines recursively. Section 3.2 provides sensitivity bound for lines-sets clustering, by translating the input lines into robust medians, converting the translated lines into colored points, and then compute sensitives for colored-sets as in Section 3.1.

Definition 3.1 (Line-set Sensitivity)**.** *Let $\mathcal{L}$ be an (n, m)-set of lines in $\mathbb{R}^d$, and let $k \geq 1$ be an integer. We define the* sensitivity *of every such set $L \in \mathcal{L}$ of lines to be*

$$S_{\mathcal{L},k}(L) = \sup_{C \subseteq \mathbb{R}^d, |C| = k} \frac{\tilde{D}(L, C)}{\sum_{L' \in \mathcal{L}} \tilde{D}(L', C)},$$

where the supremum is over every set C of k points in $\mathbb{R}^d$ such that $\tilde{D}(L, C) > 0$.

3.1 Colored sets

In this subsection, we bound the sensitivity for the *colored sets clustering*. Here, the input is a set $\mathcal{P}$ of n m-ordered-sets (the order resembles the colors of the points) in a metric space $(\mathcal{X}, d)$ equipped with a distance function $\tilde{D}$, and an integer $k \geq 1$. The goal is to compute a colored set $\subseteq \mathcal{X} \times [m]$ of k pairs, each consisting of a point and a index (color) in $[m]$ that minimizes the sum of distances to the sets. Here, the distance to each ordered-set $P \in \mathcal{P}$ is the minimum distance from a center in to a point in P which has the same color,

$$\in \operatorname*{arg\,min}_{C' \subseteq \mathcal{X} \times [m], |C'|=k} \sum_{(p_1,\ldots,p_m) \in \mathcal{P}} \min_{(c,t) \in C'} D(c, p_t).$$

Example. Suppose we want to provide a convenient food source for n workers whose home and work addresses are represented by n pairs of GPS coordinates (points on the plane). Each person can either buy ingredients at the grocery and make a lunch box at home, or go to a restaurant during lunchtime; see Fig 3.

In order to make the reduction to sets of lines, we need coreset for this problem that supports weights. Hence, we define the weighted version of this problem.

Definition 3.2 (weighted colored center)**.** *A weighted colored center in $\mathcal{X}$ is a triplet (c, w, t), where $c \in \mathcal{X}$, $w \geq 0$, and $t \geq 1$ is an integer. We define the weighted distance from an m-ordered set $P = (p_1, \ldots, p_m)$ to a colored weighted center $c' = (c, w, t)$ to be*

$$\tilde{D}(P, c') = \tilde{D}(P, (c, w, t)) = \begin{cases} w \cdot \tilde{D}(p_t, c), & \text{if } t \leq m \\ \infty, & \text{otherwise} \end{cases}.$$

The distance between a finite set C of weighted colored centers, and an m-ordered-set P is defined by $\tilde{D}(P, C) = \min_{c' \in C} \tilde{D}(P, c')$, and the cost between C and an (n, m)-ordered set $\mathcal{P}$ is the sum $\tilde{D}(\mathcal{P}, C) = \sum_{P \in \mathcal{P}} \tilde{D}(P, C)$.

Overview of Algorithm 1. Given a set $\mathcal{P}$ of m-ordered-sets and an integer $k \geq 1$, Algorithm 1 computes a set $\mathcal{B}_k^m$ of center points (used only for the analyses) and a set $\mathcal{P}_k^m \subseteq \mathcal{P}$ of "similar" m-ordered-sets, which are approximately equally important for the problem at hand; see Lemma 3.3. On the ℓ-th iteration of the external loop (Line 2), the algorithm computes a fraction of $O(\frac{1}{k^m})$ m-sets from $\mathcal{P}_{\ell-1}^m$ that are similar in the sense that there are m dense balls of small radii, each contains at least one point from each set.

Algorithm 1: CS-DENSE$(\mathcal{P}, k)$

Input : An (n, m)-ordered-set $\mathcal{P}$ in $\mathcal{X}$, and an integer $k \geq 1$.
Output : A pair $(\mathcal{P}_k^m, \mathcal{B}_k^m)$, where $\mathcal{P}_k^m \subseteq \mathcal{P}$ and $\mathcal{B}_k^m \subseteq \mathcal{X} \times [m]$; see Lemma 3.3.

1 $\mathcal{P}_1^0 := \mathcal{P}$; $\mathcal{B}_1^0 := \emptyset$; $\tau := \dfrac{1}{20}$
2 **for** $r := 1$ **to** k **do**
3   **for** $\ell := 1$ **to** m **do**
4         $\overline{\mathcal{P}}_r^{\ell-1} := \left\{ \overline{\text{proj}}(P, \mathcal{B}_r^{\ell-1}) \middle| P \in \mathcal{P}_r^{\ell-1} \right\}$ `// see Definition 2.7`
5   Compute a $\left(\dfrac{1}{2k}, \tau, 2\right)$-median $b_r^\ell \in \mathcal{X} \times [m - (\ell - 1)]$ for $\overline{\mathcal{P}}_r^{\ell-1}$
        `// see Definition 2.5, and suggested implementation in Algorithm 6.`
6         $\mathcal{P}_r^\ell := \left\{ P \in \mathcal{P}_r^{\ell-1} \;\middle|\; \overline{\text{proj}}(P, \mathcal{B}_r^{\ell-1}) \in \text{closest}\left(\overline{\mathcal{P}}_r^{\ell-1}, \{b_r^\ell\}, \frac{1-\tau}{4k}\right) \right\}$
        `//` $\mathcal{P}_r^\ell$ `contains every` m`-set` P `such that` $\overline{\text{proj}}(P, \mathcal{B}_r^{\ell-1})$ `is in the fraction of the closest` $(1-\tau)/(4k)$ `sets in` $\overline{\mathcal{P}}^{\ell-1}$ `to the center` b_r^ℓ`; see Definition 2.7.`
7         $\mathcal{B}_r^\ell := \mathcal{B}_r^{\ell-1} \bigcup \{b_r^\ell\}$
8     $\mathcal{P}_{r+1}^0 := \mathcal{P}_r^m$; $\mathcal{B}_{r+1}^0 := \mathcal{B}_r^m$
9 **Return** $(\mathcal{P}_k^m, \mathcal{B}_k^m)$

Lemma 3.3 (sensitivity of colored-sets)**.** *Let $\mathcal{P}$ be an (n, m)-ordered-set in $\mathcal{X}$, $k \geq 1$ be an integer, and let $(\mathcal{P}_k^m, \mathcal{B}_k^m)$ be the output of a call to* CS-DENSE$(\mathcal{P}, k)$*; see Algorithm 1. Then $|\mathcal{P}_k^m| \in \Theta(n)$, and for every set $P \in \mathcal{P}_k^m$ and a set $C \subseteq \mathcal{X} \times [0, \infty) \times [m]$ of $|C| = k$ weighted colored centers such that $\tilde{D}(P, C) > 0$, we have $\frac{\tilde{D}(P,C)}{\sum_{P' \in \mathcal{P}} \tilde{D}(P',C)} \leq \frac{2k}{|\mathcal{P}_k^m|}$.*

3.2 Line-sets

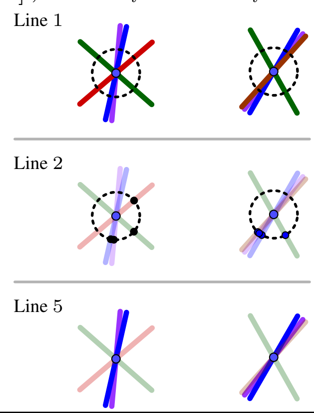
Overview of Algorithm 2. Given an (m, m)-set $\mathcal{L}$ of lines, and a set $B \subseteq \mathbb{R}^d$ that both satisfy the condition of Lemma C.4, and an integer $k \geq 1$. Algorithm 2 outputs a function $s : \mathcal{L} \to [0, \infty)$ that maps every set $L \in \mathcal{L}$ of lines to an upper bound $s(L)$ of its sensitivity. This is done by a reduction to colored-sets clustering. Intuitively, the distance between a projected set of lines (see Fig. 2 (a)) and a center can be approximated by a distance between a colored set and a set of colored centers (projections of the original center) as shown in Fig. 2(b)

Algorithm 2: $\textsc{Grouped-Sensitivity}(\mathcal{L}, B, k)$

Input : An integer $k \geq 1$, an (n, m)-set $\mathcal{L}$ of lines, and a set $B = (b_1, \ldots, b_m)$ of m points both in $\mathbb{R}^d$ such that for every set $L = (\ell_1, \ldots, \ell_m) \in \mathcal{L}$ and every $i \in [m]$, the line ℓ_i intersects b_i.

Output : A sensitivity function $s : \mathcal{L} \to [0, \infty)$, for every set $L \in \mathcal{L}$.

1 For every $i \in [m]$, set $\mathbb{S}_i := \left\{ x \in \mathbb{R}^d \mid \|x - b_i\| = 1 \right\}$
2 For every $L \in \mathcal{L}$, set $P(L) := (\ell_i \cap S_i)_{i=1}^{m}$ `// A` $2m$`-ordered set`
3 $\mathcal{P} := \left\{ P(L) \middle| L \in \mathcal{L} \right\}$ `// An` $(n, 2m)$`-ordered set`
4 **while** $|\mathcal{P}| > 2mk$ **do**
5 $\quad (\mathcal{P}_k^m, \mathcal{B}_k^m) := \textsc{CS-Dense}(\mathcal{P}, mk)$ `// see Algorithm 1`
6 $\quad$ **for** *every* $P \in \mathcal{P}^m$ **do**
7 $\quad\quad s'(P) := \dfrac{2mk}{|\mathcal{P}_k^m|}$ `// see Lemma 3.3`
8 $\quad \mathcal{P} := \mathcal{P} \setminus \mathcal{P}_k^m$
9 **for** *every* $Q \in \mathcal{P}$ **do**
10 $\quad s'(Q) := 1$
11 **for** every $L \in \mathcal{L}$ **do** $s(L) := \sqrt{2} s'(P(L))$.
12 **Return** s



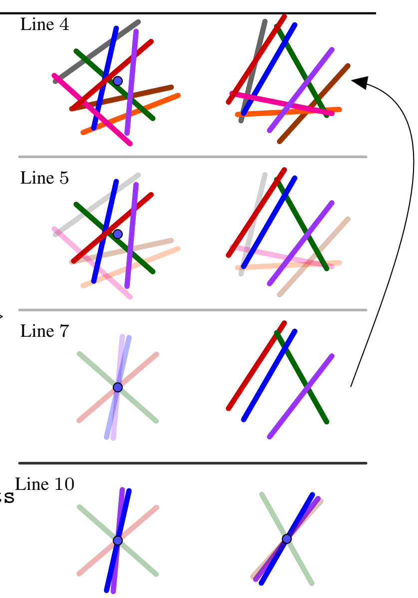
Overview of Algorithm 3 Given an (n, m)-set $\mathcal{L}$ of lines, and an integer $k \geq 1$, Algorithm 3 outputs a pair of sets $(\mathcal{L}^{m+1}, \mathcal{B}^m)$. All the m-sets in $\mathcal{L}^{m+1}$ are similar in the sense that their sensitivity with respect to original problem is small and equal. The algorithm consists of two parts. The first part (Lines 3-8), extracts a subset of the original data that can be "grouped" (see Fig. 2(a)) with little effect on the sensitivity related to the original set. The second part (Lines 9-10), uses Algorithm 2 to extract the biggest subset of this subset whose sensitivity can be bounded.

Algorithm 3: $\textsc{LS-Dense}(\mathcal{L}, k)$

Input : An (n, m)-set $\mathcal{L}$ and an integer $k \geq 1$.

Output : A pair $(\mathcal{L}^m, \mathcal{B}^m)$, where $\mathcal{L}^m \subseteq \mathcal{L}$ and $\mathcal{B}^m \subseteq \mathbb{R}^d$ is an ordered set.

1 $\tau := \frac{1}{20}$
2 $\bar{\mathcal{L}}^0 := \mathcal{L}^0 := \mathcal{L}$; $\mathcal{B}^0 := \emptyset$
3 **for** $i := 1$ **to** m **do**
4 $\quad$ Compute a $\left(\dfrac{1}{2k}, \tau, 4\right)$-median $b^i \in \mathbb{R}^d$ for $\bar{\mathcal{L}}^{i-1}$
$\quad$ `// see Algorithm 6 for suggested implementation`
5 $\quad \mathcal{L}^i := \left\{ L \in \mathcal{L}^{i-1} \;\middle|\; \overline{\text{proj}}(L, \mathcal{B}^{i-1}) \in \text{closest}\left(\bar{\mathcal{L}}^{i-1}, \{b^i\}, \frac{1-\tau}{4k}\right) \right\}$
6 $\quad \mathcal{B}^i := \mathcal{B}^{i-1} \bigcup \{b^i\}$
7 $\quad \bar{\mathcal{L}}^{i-1} := \left\{ \overline{\text{proj}}(L, \mathcal{B}^{i-1}) \middle| L \in \mathcal{L}^{i-1} \right\}$ `// see Fig 3`
8 $\mathcal{L}' := \left\{ \text{proj}(L, \mathcal{B}^m) \middle| L \in \mathcal{L}^m \right\}$
9 $s := \textsc{Grouped-Sensitivity}(\mathcal{L}', \mathcal{B}^m, k)$ `// see Algorithm 2`
10 $\mathcal{L}^{m+1} := \underset{L \in \mathcal{L}^m}{\arg\min}\, s\Big(\text{proj}(L, \mathcal{B}^m)\Big)$ `//` $\mathcal{L}^{m+1}$ `is a set of sets`
`// biggest cluster whose sets have equal sensitivity`
11 **Return** $(\mathcal{L}^{m+1}, \mathcal{B}^m)$



Lemma 3.4. *Let $\mathcal{L}$ be an (n, m)-set of lines in $\mathbb{R}^d$, and $k \geq 1$ be an integer. Let $\left(\mathcal{L}^{m+1}, \mathcal{B}^m\right)$ be the output of a call to* $\textsc{LS-Dense}(\mathcal{L}, k)$*; see Algorithm 3. Then, for every $L \in \mathcal{L}^{m+1}$, we have* $S_{\mathcal{L},k}(L) \in O(k) \cdot \left(\frac{1}{|\mathcal{L}^{m+1}|}\right)$.

4 From Sensitivity to Coreset

As previously stated, there are many frameworks (such as [14]) which provide a coreset given an upper bound of the sensitivity as in Lemma 3.3 and an upper bound for the VC-dimension as given in Section B.In what follows we present a general algorithm that uses such a framework.

Overview of Algorithm 4 Given an (n, m)-set $\mathcal{L}$ of lines in $\mathbb{R}^d$, an integer $k \geq 1$, an error parameter $\varepsilon \in (0, 1)$, and a probability of failure $\delta \in (0, 1)$, Algorithm 4 computes an ε-coreset $(\mathcal{C}, v)$ for $\mathcal{L}$. First, the algorithm (Lines 1-7) recursively extracts a subset with known sensitivities using LS-DENSE until all sets are assigned a sensitivity, then calls an existing framework to compute the coreset (Line 8).

Algorithm 4: CORESET$(\mathcal{L}, k, \eta)$; see Theorem 4.1

Input : An (n, m)-set $\mathcal{L}$ of lines in $\mathbb{R}^d$, a positive integer $k \geq 1$, desired coreset size $\eta \geq 1$.
Output : A pair $(\mathcal{C}, v)$, where $\mathcal{C} \subseteq \mathcal{L}$ and $v : \mathcal{C} \to (0, \infty)$.

1 $\mathcal{L}^0 := \mathcal{L}^{m+1} := \mathcal{L}$
2 **while** $|\mathcal{L}^{m+1}| \geq 2$ **do**
3 $\quad (\mathcal{L}^{m+1}, \mathcal{B}^m) :=$ LS-DENSE$(\mathcal{L}^0, k)$`// see Algorithm 3`
4 $\quad$ **for** *every* $L \in \mathcal{L}^{m+1}$ **do**
5 $\quad\quad s(L) := \dfrac{1}{|\mathcal{L}^{m+1}|}$
6 $\quad \mathcal{L}^0 := \mathcal{L}^0 \setminus \mathcal{L}^{m+1}$
7 **for** every set $L \in \mathcal{L}^0$ **do** $s(L) := 1$
8 $(\mathcal{C}, v) :=$ CORESET-FW$(\mathcal{L}, s, \eta)$`// see Algorithm 5`
9 **Return**$(\mathcal{C}, v)$

Theorem 4.1. *Let $\mathcal{L}$ be an (n, m)-set of lines in $\mathbb{R}^d$, $k \geq 1$ be an integer, $\varepsilon, \delta \in (0, 1)$, and let $\eta \geq \left(\frac{m^{1.5} d \log n}{\varepsilon}\right)^2 (2k)^{cmk} + \log_2\left(\frac{1}{\delta}\right)$ be an integer, where c is a sufficiently large constant that can be determined from the proof. Let $(\mathcal{C}, v)$ be the output of a call to* CORESET$(\mathcal{L}, k, \eta)$*; see Algorithm 4 in the Appendix. Then, Claims (i)–(ii) hold as follows.*

(i) *With probability at least $1 - \delta$, $(\mathcal{C}, v)$ is an ε-coreset of $\mathcal{L}$ for the k-line-sets of size $|\mathcal{C}| = \eta$.*
(ii) *The pair $(\mathcal{C}, v)$ can be computed in $n \log(n)(2k)^{O(mk)}$ time.*

By modifying Line 3 of Algorithm 4 to call CS-DENSE rather than LS-DENSE, as well as straightforward modifications over the input, one can achieve coreset construction for Colored-sets clustering as follows.

Theorem 4.2. *Let $k, m \geq 1$ be constant integers, let $\mathcal{P}$ be an (n, m)-ordered-set in $\mathbb{R}^d$, $\varepsilon, \delta \in (0, 1)$, and let $\eta \geq \left(\frac{m^{1.5} d \log n}{\varepsilon}\right)^2 (2k)^{cmk} + \log_2\left(\frac{1}{\delta}\right)$ be an integer, where c is a sufficiently large constant that can be determined from the proof. There is an algorithm that given $\mathcal{P}, k, \varepsilon$ and δ returns with probability at least $1 - \delta$ an ε-coreset of $\mathcal{P}$ for colored-sets k-mean of size η in $n \log(n)(2k)^{O(mk)}$ time.*

5 Experimental Results

We implemented our coreset construction algorithms for both Colored-sets and Line-sets. In this section we test there empirical performance on synthetic and real data. Open source is available in [32]. Since we provide the first coreset construction for the given problems, we compare our results to a single baseline - a random uniform sampling of the same number of points as in the coreset. Except for the case of $(n, 1)$-sets of lines where we also compare to the existing coreset for lines k-mean [35].

Colored-sets data-sets

(i) A *synthetic dataset* of size $n = 10000$ with outliers by sampling colored points from a set of Gaussians with different parameters corresponding to different colors. For each color there are two overlapping Gaussians for modeling the inliers of size 4950 each and one Gaussian more than three standard deviations away of size 100 for modeling the outliers.

(ii) The Reuters-21578 data-set from [30], which results in sets of points corresponding to each paragraph in each document. Additionally we divide paragraphs into categories: beginning (first third), main part (middle third) ending (last third). This results in each document being represented as a set of m=3 colored points corresponding to sets of paragraphs of different categories.

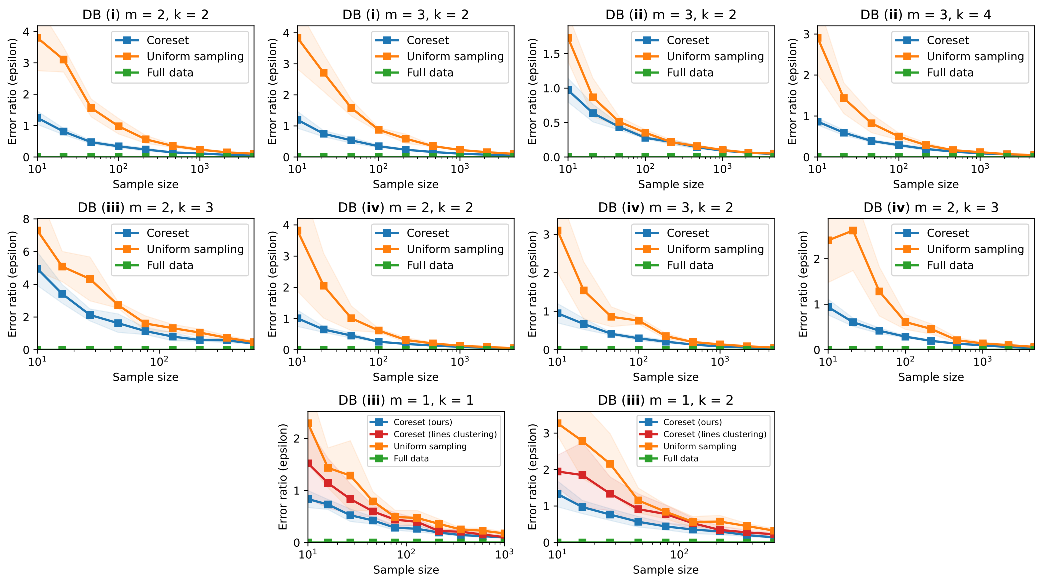


Figure 4: **Experimental results.** *See details in Section 5.*

Line-sets data-sets

(iii) A *synthetic dataset* that consists of the colored points dataset from the previous section, after removing the colors from the points. Each point was then turned into a line by assigning it to a random direction according to similar mixture of Gaussians.

(iv) We used California housing prices data-set [37] in witch we introduced uncertainty by removing two of the 9 dimensions each point. We removed one discrete value which created sets and one continuous value which created the lines.

The experiment. For each data set, and for different values of m, k, we conduct the following experiments. For every $\sigma \in \{10 \cdot 2^i\}_{i=0}^{n}$, a coreset $\mathcal{S}_1(\sigma)$ of size σ using our algorithm, a uniform sample $\mathcal{S}_2(\sigma)$ of the data-set of size σ, and for the corresponding experiments a coreset $\mathcal{S}_3(\sigma)$ of size σ using [35]. Then we generated a set $\mathcal{Q}$, $|Q| = 500$ queries: half of them are k-means that were computed using generalization of the EM heuristic, and the rest were randomly and uniformly sampled from the ground set. Finally, for every $i \in \{1, 2\}$ we computed the maximum approximation error

$$\mathcal{S}_i, \varepsilon_i(\sigma) = \max_{Q \in \mathcal{Q}} \frac{\left| \sum_{P \in \mathcal{P}} \tilde{D}(\mathcal{S}_i, Q) - \sum_{P \in \mathcal{P}} \tilde{D}(\mathcal{P}, Q) \right|}{\sum_{P \in \mathcal{P}} \tilde{D}(\mathcal{P}, Q)}$$

Results for the corresponding databases (i)–(iv) are shown in Fig. 4.

Discussion Our coresets out-preform uniform sampling in most of the experiments. As expected, when the data is uniformly distributed, the sensitivity of each point is close, and uniform sampling is already a coreset. Increasing k and m yields more isolated clusters, which explains why the error for the uniform samplings becomes higher compared to the coreset's error. As is in previous paper, the empirical error is very small compared to the theoretical worst-case bounds, even for small coresets.

6 Conclusion and Future Work

The paper suggests the first coresets for k line-sets clustering via a reduction to coresets for colored-sets clustering of independent interest. We expect that this paper will open a line of research for many possible directions. For example, projectivedd clustering of sets. That is, replacing the *outputted* k point centers by k linear subspaces of $\mathbb{R}^d$, each of dimension $j \geq 1$. Another direction is to handle inputs sets of subspaces, each of dimension $j \geq 1$. Extending to non-linear shapes is another direction. Although, the main contribution of this paper is the theoretical breakthrough and results. However, we also expect that our code can be extended and applied for real-world systems that have similar problems, such as [24, 34].

References

[1] P. K. Agarwal, G. Cormode, Z. Huang, J. M. Phillips, Z. Wei, and K. Yi. Mergeable summaries. *ACM Transactions on Database Systems (TODS)*, 38(4):1–28, 2013.

[2] S. Alipour. Approximation algorithms for probabilistic k-center clustering. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1–11, 2020.

[3] M. Anthony and P. L. Bartlett. *Neural network learning: Theoretical foundations*. Cambridge University press, 2009.

[4] D. Arthur and S. Vassilvitskii. How slow is the k-means method? In *Proceedings of the twenty-second annual symposium on Computational Geometry (SoCG)*, pages 144–153, 2006.

[5] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[6] H. Azzag, G. Venturini, A. Oliver, and C. Guinot. A hierarchical ant based clustering algorithm and its use in three real-world applications. *European Journal of Operational Research*, 179(3):906–922, 2007.

[7] A. Barger and D. Feldman. Deterministic coresets for k-means of big sparse data. *Algorithms*, 13(4):92, 2020.

[8] F. Bartolozzi, A. de Gaetano, E. Di Lena, S. Marino, L. Nieddu, and G. Patrizi. Operational research techniques in medical treatment and diagnosis: A review. *European Journal of Operational Research*, 121(3):435–466, 2000.

[9] V. Braverman, D. Feldman, and H. Lang. New frameworks for offline and streaming coreset constructions. *arXiv preprint arXiv:1612.00889*, 2016.

[10] V. Braverman, S. Jiang, R. Krauthgamer, and X. Wu. Coresets for clustering with missing values. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:17360–17372, 2021.

[11] V. Braverman, S. H.-C. Jiang, R. Krauthgamer, and X. Wu. Coresets for ordered weighted clustering. In *International Conference on Machine Learning (ICML)*, pages 744–753. PMLR, 2019.

[12] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii. Fair clustering through fairlets. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30. Curran Associates, Inc., 2017.

[13] V. Cohen-Addad, D. Saulpic, and C. Schwiegelshohn. A new coreset framework for clustering. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 169–182, 2021.

[14] D. Feldman, Z. Kfir, and X. Wu. Coresets for gaussian mixture models of any shape, 2019.

[15] D. Feldman and M. Langberg. A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM symposium on Theory of computing (STOC)*, pages 569–578, 2011.

[16] D. Feldman and L. J. Schulman. Data reduction for weighted and outlier-resistant clustering. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1343–1354. SIAM, 2012.

[17] J. Gao, M. Langberg, and L. J. Schulman. Clustering lines in high-dimensional space: Classification of incomplete data. *ACM Trans. Algorithms*, 7(1), Dec. 2010.

[18] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *2011 IEEE intelligent vehicles symposium (IV)*, pages 963–968. IEEE, 2011.

[19] S. Har-Peled. Coresets for discrete integration and clustering. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 33–44. Springer, 2006.

[20] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.

[21] L. Huang and N. K. Vishnoi. Coresets for clustering in euclidean spaces: Importance sampling is nearly optimal. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1416–1429, 2020.

[22] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011.

[23] M. Joseph, M. Kearns, J. Morgenstern, and A. Roth. Fairness in learning: Classic and contextual bandits. *arXiv preprint arXiv:1605.07139*, 2016.

[24] I. Jubran, F. Fares, Y. Alfassi, F. Ayoub, and D. Feldman. Newton-pnp: Real-time visual navigation for autonomous toy-drones. *arXiv preprint arXiv:2203.02686*, 2022.

[25] I. Jubran, E. E. S. Shayda, I. Newman, and D. Feldman. Coresets for decision trees of signals. In *Thirty-Fifth Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[26] I. Jubran, M. Tukan, A. Maalouf, and D. Feldman. Sets clustering. *International Conference on Machine Learning (ICML)*, 2020.

[27] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer, 2012.

[28] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. A local search approximation algorithm for k-means clustering. *Computational Geometry*, 28(2-3):89–112, 2004.

[29] M. Langberg and L. J. Schulman. Universal ε-approximators for integrals. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms (SODA)*, pages 598–607. SIAM, 2010.

[30] D. Lewis. Reuters-21578 text categorization test collection, distribution 1.0. *http://www.research.att.com*, 1997.

[31] J. Li, W. Wu, T. Wang, and Y. Zhang. One step beyond histograms: Image representation using markov stationary features. In *2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[32] S. Lotan, E. E. S. Shayda, and D. Feldman. Coreset for lines sets clustering - open source code. https://github.com/ernestosanches/Sets-of-lines-clustering, 2022.

[33] M. Lucic, O. Bachem, and A. Krause. Strong coresets for hard and soft bregman clustering with applications to exponential family mixtures. In *Artificial intelligence and statistics (AISTAT)*, pages 1–9. PMLR, 2016.

[34] A. Maalouf, I. Jubran, and D. Feldman. Fast and accurate least-mean-squares solvers. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[35] Y. Marom and D. Feldman. k-means clustering of lines for big data. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 12817–12826, 2019.

[36] R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The effectiveness of lloyd-type methods for the k-means problem. *Journal of the ACM (JACM)*, 59(6):1–22, 2013.

[37] R. K. Pace and R. Barry. California housing dataset. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html, 1997.

[38] G. Rosman, M. Volkov, D. Feldman, J. W. Fisher III, and D. Rus. Coresets for k-segmentation of streaming data. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27. Curran Associates, Inc., 2014.

[39] M. Schmidt, C. Schwiegelshohn, and C. Sohler. Fair coresets and streaming algorithms for fair k-means. In *International Workshop on Approximation and Online Algorithms*, pages 232–251. Springer, 2019.

[40] X. Shen, W. Liu, I. Tsang, F. Shen, and Q.-S. Sun. Compressed k-means for large-scale clustering. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[41] C. Sohler and D. P. Woodruff. Strong coresets for k-median and subspace approximation: Goodbye dimension. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 802–813. IEEE, 2018.

[42] A. Statman, L. Rozenberg, and D. Feldman. k-means+++: Outliers-resistant clustering. *Algorithms*, 13:311, 11 2020.

[43] A. Vattani. K-means requires exponentially many iterations even in the plane. *Discrete & Computational Geometry*, 45(4):596–616, 2011.

A Appendix

B Sensitivity Coresets

Every coreset in this paper is a weighted subset of its input, unlike other papers where the coreset may be a subset of a larger ground set. For example, points or lines in $\mathbb{R}^d$ instead of the input set. For clarity, the following definitions are a bit simpler than their cited versions due to this feature of our coresets. We also assume that the set of queries is the same for every input set and not a function of the resulting coreset. Such a generalization may be used to remove the dependency of the coreset on the dimension d; see Future Work in Section 6.

Definition B.1 (Query space [9, 29])**.** *Let P be a finite set. Let Q be a (possibly infinite) set, called* queries*. Let $f : P \times Q \to [0, \infty)$ be a cost function. The tuple (P, Q, f) is called a* query space*.*

In most papers, P is a set of points in $\mathbb{R}^d$ or a metric space, and in some papers, P is a set of lines. However, in this paper, P is usually a set of sets, where each $p \in P$ corresponds to a set of lines or colored points. Every item in Q is a set of k centers, sometimes colored. Since this section is generic, we keep the classic notation defined above.

The following definition of VC-dimension is used in Theorem B.5 to bound the VC-dimension for our problems. Unfortunately, a different definition is used in the context of query spaces. Below we present the two definitions and show how they are equivalent.

Definition B.2 (VC-dimension)**.** *Let $\mathcal{X}$ be a set, called* ground set*, and let $\mathcal{F}$ be a set of functions from $\mathcal{X}$ to $\{0, 1\}$. For a set $S \subseteq \mathcal{X}$, we call $S_f := \{x \in S | f(x) = 1\}$ the subset of S* induced *by f. We say that S is* shuttered *by $\mathcal{F}$ if and only if $|\{S_f | f \in \mathcal{F}\}| = 2^{|S|}$. The VC-dimension of $\mathcal{F}$ is the largest size of S that is shuttered by $\mathcal{F}$.*

The definition of VC-dimension for query spaces is a straight forward generalization of the classic definition of VC-dimension in PAC-learning above.

Definition B.3 (VC-dimension for query spaces [9])**.** *For a query space (P, Q, f), a query $C \in Q$, and $r \in [0, \infty)$ we define*

$$\mathrm{range}(P, C, r) = \{p \in P | f(p, C) \leq r\}.$$

Let $\mathrm{ranges}(P, Q, f) := \left\{\mathrm{range}(P, C, r) \middle| C \in Q, r \geq 0\right\}$. *The VC-dimension of the pair $(P, \mathrm{ranges}(P, Q, f))$ is the size $|S|$ of the largest subset $S \subseteq P$ such that*

$$|\left\{S \cap \mathrm{range}(P, C, r) \middle| C \in Q, r \in [0, \infty)\right\}| = 2^{|S|}.$$

The VC-dimension of the query space (P, Q, f) *is the VC-dimension of* $(P, \mathrm{ranges}(P, Q, f))$.

The following lemma show the relation between the pair of definitions above.

Lemma B.4. *Let (P, Q, f) be a query space. We define*

$$\mathcal{H}((P, Q, f)) := \left\{ x \mapsto \begin{cases} 1, & f(x, C) \leq r \\ 0, & \text{otherwise} \end{cases} \middle| C \in Q, r \in [0, \infty) \right\}$$

The VC-dimension of $\mathcal{H}((P, Q, f))$ is the VC-dimension of (P, Q, f).

Proof. Let $H = \mathcal{H}(P, Q, f)$ be as defined in Lemma B.4. Let $S \subseteq P$ be a set, and for every $h \in H$ let S_h be the subset of S induced by h. For every $C \in Q$ and $r \in [0, \infty)$, let $h'(x) = \begin{cases} 1, & f(x, C) \leq r \\ 0, & \text{otherwise} \end{cases}$ then every $x \in P$ is $x \in S_{h'} \Leftrightarrow x \in S \cap \mathrm{range}(P, C, r)$. Hence, for every $S' \subseteq S$ we have $S' \in \{S_h | h \in H\} \Leftrightarrow S' \in \left\{S \cap \mathrm{range}(P, C, r) \middle| C \in Q, r \in [0, \infty)\right\}$. Finally, the VC-dimension of both, H and (P, Q, f) is the size of largest set S, such that $|\{S_h | h \in H\}| = \left|\left\{S \cap \mathrm{range}(P, C, r) \middle| C \in Q, r \in [0, \infty)\right\}\right| = 2^{|S|}$. □

Lemma B.5 (Variant of Theorem 8.4, [3])**.** *Suppose h is a function from $\mathbb{R}^d \times \mathbb{R}^n$ to $\{0, 1\}$ and let*

$$H = \left\{x \mapsto h(a, x) \middle| a \in \mathbb{R}^d, x \in \mathbb{R}^n\right\}$$

be the class determined by h. Suppose that h can be computed by an algorithm that takes as an input a pair $(a, x) \in \mathbb{R}^d \times \mathbb{R}^n$ and returns $h(a, x)$ after no more than t operations of the following types:

- *the arithmetic operations* $+, -, \times$, *and* $/$ *on real numbers,*
- *jumps conditioned on* $>, \geq, <, \leq, =$, *and* $\neq$ *comparisons of real numbers, and*
- *outputs* 0 *or* 1.

Then the VC*-dimension of* H *is* $O\left(dt\right)$.

We now bound the VC-dimension for our line-sets mean problem, whose query space is $\left(\mathcal{L}, \{C \subseteq \mathbb{R}^d||C| = k\}, \tilde{D}\right)$.

Lemma B.6. *Let* $\mathcal{L}$ *be an* (n, m)*-set of lines in* $\mathbb{R}^d$, $k \geq 1$ *be an integer, and* $Q = \{C \subseteq \mathbb{R}^d||C| = k\}$. *Let* $(\mathcal{L}, Q, \tilde{D})$ *be a line-sets clustering query space. Then the VC-dimension* d' *of* $(\mathcal{L}, Q, \tilde{D})$ *is* $d' \in O(md^2k^2)$.

Proof. Let $\mathcal{Q} = \{C \subseteq \mathbb{R}^d\big||C| = k\}$. Let $H = \mathcal{H}((\mathcal{L}, \mathcal{Q}, \tilde{D}))$ be as defined in Lemma B.4. Let $h : \mathcal{L} \times (Q \times \mathbb{R}) \to \{0, 1\}$ such that

$$h(L, (C, r)) = \begin{cases} 1, & if\ \tilde{D}(L, C) \geq r \\ 0, & Otherwise. \end{cases}$$

Note that, H is determined by h. Assuming that each line $\ell \in L \in \mathcal{L}$ is represented by direction vector, it takes $t = O(mdk)$ arithmetic operations to evaluate h. Furthermore, any pair in $\mathcal{Q} \times \mathbb{R}$ can be represented as a vector in $(dk + 1)$-dimensional space. Hence by Lemma B.5, the VC-dimension of H is $O(dk \cdot mdk) = O(md^2k^2)$. Finally, by Lemma B.4 the VC-dimension of $(\mathcal{L}, \mathcal{Q}, \tilde{D})$ is equal to the VC-dimension of H and both are $O(md^2k^2)$. $\square$

In this paper we use the classic definition of sensitivity.

Definition B.7 (Sensitivity)**.** *Let* (P, Q, f) *be a query space. The* sensitivity *function* $s^* : P \to [0, \infty)$ *of a query space* (P, Q, f) *maps every* $p \in P$ *to*

$$s^*(p) := \sup_q \frac{f(p, q)}{\sum_{p' \in P} f(p', q)},$$

where the supremum is over every $q \in Q$ *such that* $f(p, q) > 0$.

An upper bound for the sensitivity of such a query space is a function $s : P \to [0, \infty)$ *such that* $s(p) \geq s^*(p)$ *for every* $p \in P$.

In the above definition we assumed that the supremum of an empty set is zero.
The following theorem proves that a coreset can be computed by sampling according to sensitivity of points. The size of the coreset depends on the total sensitivity and the complexity (VC-dimension) of the query space, as well as the desired error ε and probability δ of failure.

Theorem B.8 ([9])**.** *Let*

- (P, Q, f) *be a query space, and* $n = |P|$*.*
- d' *be the dimension of* (P, Q, f)*.*
- $s : P \to [0, \infty)$ *be a sensitivity bound of* (P, Q, f)*, and* $t = \sum_{p \in P} s(p)$ *be its total sensitivity.*
- $\varepsilon, \delta \in (0, 1)$*,*
- $c > 0$ *be a universal constant that can be determined from the proof,*
- $\eta \geq \frac{c(t+1)}{\varepsilon^2} \left(d' \log(t+1) + \log\left(\frac{1}{\delta}\right) \right)$ *, and*
- $(\mathcal{C}, v)$ *be the output weighted set of a call to* CORESET-FW(P, s, η)*; see Algorithm 5.*

Then (i)*–*(ii) *hold as follows.*

(i) With probability at least $1 - \delta$*,* $(\mathcal{C}, v)$ *is an* ε*-coreset of* (P, Q, f)*, whose size is* $|\mathcal{C}| = \eta$*; see Section 1.2.*

(ii) $(\mathcal{C}, v)$ *can be computed in* $O(n)$ *time, given* (P, s, η)*.*

Algorithm 5: CORESET-FW(P, s, η)

Input :A finite set $P \subseteq \mathbb{R}^d$, where $\sum_{p \in P} w(p) > 0$, a function $s : P \to [0, \infty)$, and an integer $\eta \geq 1$.
Output :A weighted set (C, v).

$s'(p) := s(p) + \frac{1}{|P|}$
$\mathcal{C} := \left\{ p \in P \middle| \frac{s'(p)}{\sum_{q \in P} s'(q)} \geq \frac{1}{\eta} \right\}$ **for** every $p \in \mathcal{C}$ **do** $v'(p) := 1$
$Q := P \setminus \mathcal{C}$
$\eta' := \eta - |\mathcal{C}|$
for η' *iterations* **do**
  Sample a point q from Q, where $q = p$ with probability $\Pr(p) := \dfrac{s'(p)}{\sum_{p' \in Q} s'(p')}$ $\mathcal{C} := \mathcal{C} \cup \{q\}$
  $v'(q) := \dfrac{1}{\eta' \cdot \mathrm{pr}(q)}$
for *every* $p \in \mathcal{C}$ **do**
  $v(p) := v'(p) \cdot \dfrac{|P|}{\sum_{q \in \mathcal{C}} v'(q)}$
Return (C, v)

Theorem B.9 (Restatement of Theorem 4.1)**.** *Let* $\mathcal{L}$ *be an* (n, m)*-set of lines in* $\mathbb{R}^d$*,* $k \geq 1$ *be an integer,* $\varepsilon, \delta \in (0, 1)$*, and let*

$$\eta \geq \left(\frac{m^{1.5} d \log n}{\varepsilon} \right)^2 (2k)^{cmk} + \log\left(\frac{1}{\delta}\right)$$

be an integer, where c *is sufficiently large constant that can be determined from the proof. Let* $(\mathcal{C}, v)$ *be the output of a call to* CORESET$(\mathcal{L}, k, \eta)$*; see Algorithm 4. Then, Claims (i)–(ii) hold as follows.*

(i) With probability at least $1 - \delta$*,* $(\mathcal{C}, v)$ *is an* ε*-coreset of* $\mathcal{L}$ *for the* k*-line-sets of size* $|\mathcal{C}| = \eta$*; see Section 1.2.*

(ii) The pair $(\mathcal{C}, v)$ *can be computed in* $n \log(n) (2k)^{O(mk)}$ *time.*

Proof. Let J denote the number of "while" iterations in Line 2 of Algorithm 4. For every $j \in [J]$, let $\mathcal{L}_j^0$, $\mathcal{L}_j^{m+1}$ and $\mathcal{B}_j^m$ denote, respectively, the sets $\mathcal{L}^0$, $\mathcal{L}^{m+1}$ and $\mathcal{B}^m$ during the execution of Line 3 at the jth "while" iteration.

Let $j \in [J]$. The pair $(\mathcal{L}_j^{m+1}, \mathcal{B}_j^m)$ is the output of a call to LS-DENSE$(\mathcal{L}_j^0, k)$. Hence, by Lemma 3.4, with an appropriate choice of $b \in O(k)$ (determined from the proof of Lemma 3.4), for every $L \in \mathcal{L}_j^{m+1}$ its value $s(L)$ that is defined in Lines 5 satisfies, for every $C \subseteq \mathbb{R}^d$ of size $|C| = k$, such that $\tilde{D}(L,C) > 0$,

$$b \cdot s(L) = \frac{b}{\left|\mathcal{L}_j^{m+1}\right|} \geq \frac{\tilde{D}(L,C)}{\sum\limits_{Q \in \mathcal{L}_j^0} \tilde{D}(Q,C)} \geq \frac{\tilde{D}(L,C)}{\sum\limits_{Q \in \mathcal{L}} \tilde{D}(Q,C)}, \tag{5}$$

where the first inequality is by Lemma 3.4, and the second inequality holds since $\mathcal{L}_j^0 \subseteq \mathcal{L}$. Since (5) holds for every C, and $L \in \mathcal{L}$ we have

$$b \cdot s(L) \geq \sup_{\substack{C \subseteq \mathbb{R}^d, |C|=k \\ \tilde{D}(L,C)>0}} \frac{\tilde{D}(L,C)}{\sum_{Q \in \mathcal{L}} \tilde{D}(Q,C)} \geq S_{\mathcal{L},k}(L). \tag{6}$$

By Line 8 of Algorithm 4, the pair $(\mathcal{C}, v)$ is the output of Algorithm 5. By (6) $b \cdot s(L)$ is an upper bound to the sensitivity of $\mathcal{L}$. Note that in Line 8 we call Algorithm 5 with s and not with $b \cdot s$, however, the distribution is the same due to the scaling in Line 6 of Algorithm 5. By Theorem B.8 $(\mathcal{C}, v)$, is an ε-coreset with probability at least $1 - \delta$ if

$$\eta \geq \frac{c(t+1)}{\varepsilon^2} \left(d' \log(t+1) + \log\left(\frac{1}{\delta}\right)\right), \tag{7}$$

where c is a constant that can be determined from the proof, d' is the VC-dimension of the query space $\left(\mathcal{L}, \{X \subseteq \mathbb{R}^d \mid |X| = k\}, \tilde{D}\right)$, and $t = \sum_{L \in \mathcal{L}} s(L)$ is the total sensitivity of $\mathcal{L}$.

In order to bound the total sensitivity, given a set $\mathcal{L}'$ we first determine the size of fraction of sets from $\mathcal{L}'$ that outputted by the execution of LS-DENSE$(\mathcal{L}', k)$; see Algorithm 3. Let $\tau = \frac{1}{20}$, and $a = \frac{4}{1-\tau}$. By Line 5 of Algorithm 3, for every $i \in [m]$

$$\left|\mathcal{L}^i\right| = \left\lceil \frac{\left|\mathcal{L}^{i-1}\right|}{ak} \right\rceil. \tag{8}$$

Therefore, by induction

$$\left|\mathcal{L}^m\right| \geq \frac{\left|\mathcal{L}^0\right|}{(ak)^m}. \tag{9}$$

The argmin set in Line 10 is the set of lines that is retuned in the fist iteration of Algorithm 2. Similarly to (9), by induction over Line 6 of Algorithm 1 we have

$$\left|\mathcal{L}^{m+1}\right| \geq \frac{\left|\mathcal{L}^m\right|}{(ak)^{mk}}, \tag{10}$$

and by combining (9) with (10) we get

$$\left|\mathcal{L}^{m+1}\right| \geq \frac{\left|\mathcal{L}^0\right|}{(ak)^{mk+m}}. \tag{11}$$

Combining (11) and Line 3 of Algorithm 4 yields

$$\left|\mathcal{L}_j^{m+1}\right| \geq \frac{\left|\mathcal{L}_j^0\right|}{(ak)^{mk+m}}. \tag{12}$$

By (12) and Line 6 of Algorithm 4

$$\begin{aligned} \left|\mathcal{L}_{j+1}^0\right| &\leq \left|\mathcal{L}_j^0\right| - \left|\mathcal{L}_j^{m+1}\right| \leq \left|\mathcal{L}_j^0\right| - \frac{\left|\mathcal{L}_j^0\right|}{(ak)^{mk+m}} \\ &= \left|\mathcal{L}_j^0\right| \left(1 - \frac{1}{(ak)^{mk+m}}\right) \\ &= \left|\mathcal{L}_1^0\right| \left(1 - \frac{1}{(ak)^{mk+m}}\right)^j \\ &= n \left(1 - \frac{1}{(ak)^{mk+m}}\right)^j, \end{aligned} \tag{13}$$

where the second inequality is by (12). Combining the fact that $\left|\mathcal{L}_j^0\right| \geq 1$ with (13), we conclude that

$$J \leq (ak)^{mk+m} \log_2 n. \tag{14}$$

Therefore, the total sensitivity t is bounded by

$$\begin{aligned} t &\leq \sum_{L \in \mathcal{L}} s(L) = \sum_{j=1}^{J} \left(\sum_{L \in \mathcal{L}_j^{m+1}} \frac{b}{\left|\mathcal{L}_j^{m+1}\right|} \right) + c = \sum_{j \in [J]} \left(\left|\mathcal{L}_j^{m+1}\right| \cdot \frac{b}{\left|\mathcal{L}_j^{m+1}\right|} \right) + c \\ &= \sum_{j \in [J]} b + c = Jb + c \leq (ak)^{mk+m+1} \log_2 n, \end{aligned}$$

where $c \geq 1$ is a constant. By combining with (7), we get that the pair $(\mathcal{C}, v)$ is an ε-coresets for $(\mathcal{L}, k)$ if

$$\eta \geq \frac{(ak)^{mk+m+1} \log_2 n}{\varepsilon^2} \left(\log_2 \left((ak)^{mk+m+1} \log n \right) d' + \log_2 \left(\frac{1}{\delta} \right) \right).$$

where $d' \in O(md^2k^2)$ (by Lemma B.6) is the VC-dimension of the sets clustering query space of the lines set clustering problem.

Running time: Consider a call CS-DENSE$(\mathcal{P}, k)$ to Algorithm 1 where $\mathcal{P}$ is an (n, m)-set. For every $i \in [k], j \in [m]$ the i, jth iteration of the "for" loops takes $O\left(dn\left(\frac{1}{4k}\right)^{jm+i-1} + dk^4\right)$ time. Summing over all the mk iterations yields a total running time of $O\left(dn + dmk^5\right)$.

Consider a call LS-DENSE$(\mathcal{L}, k)$ to Algorithm 3, where $\mathcal{L}$ is an (n, m)-set of lines in $\mathbb{R}^d$. For every $i \in [m]$, the ith iteration of the "for" loop at Line 3 takes $O\left(dn\left(\frac{1}{4k}\right)^{j-1} + dk^4\right)$ time. Summing over all the m iterations yields a total running time of $O(dn + dk^4)$. Combined with the call to CS-DENSE inside the call to GROUPED-SENSITIVITY at Line 9 the overall time is $O(dn + dmk^5)$.

There are J calls to LS-DENSE$(\mathcal{L}^0, k)$ at Line 3 of Algorithm 4, which dominates the running time of this algorithm (in each of the J iterations of the "while" loop). The set $\mathcal{L}^0$ at the jth call is of size $\left|\mathcal{L}_j^0\right| \in O\left(n\left(1 - \frac{1}{4k}\right)^{i-1}\right)$. Therefore, the jth call takes $O\left(d\left|\mathcal{L}_j^0\right| + dmk^5\right)$ time. Summing this running time over every $j \in [J]$, where $J \leq (ak)^{mk+m} \log n$ by (14), yields a total running time of

$$J \cdot dmk^5 + dn \sum_{i=1}^{J} \left(1 - \frac{1}{4k}\right)^{i-1} \in dn \log_2(n)(ak)^{o(mk)},$$

as claimed in (ii). □

Theorem B.10 (Restatement of Thorem 4.2)**.** *Let $\mathcal{P}$ be an (n, m)-ordered-set in $\mathcal{X}$, let $k \geq 1$ be an integer, $\varepsilon, \delta \in (0, 1)$, and let*

$$\eta \geq \left(\frac{m^{1.5} d \log n}{\varepsilon} \right)^2 (2k)^{cmk} + \log\left(\frac{1}{\delta}\right)$$

be an integer, where c is sufficiently large constant that can be determined from the proof. There is an algorithm that given $\mathcal{P}, k, \varepsilon$ and δ return with probability at least $1 - \delta$ an ε-coreset of $\mathcal{P}$ for colored-sets k-mean of size η in $n \log(n)(2k)^{O(mk)}$ time.

Such algorithm is achieved by little variation over Algorithm 4 and using CS-DENSE instead LS-DENSE.

Proof. Similar and can be deduced from the proof of Theorem B.9. □

C Algorithms Correctness

C.1 Correctness of Algorithm 1

The goal of Algorithm 1 is to find a small family (set) $\mathcal{P}'$ of $\Theta(n)$ sets that are sufficiently mutually close that they can be replaced with multiple copies of the same set with little to no effect on the cost.

In that case, all of their sensitivities would be $\sim \frac{1}{|\mathcal{P}'|}$. The proof of this lemma is by case analesis of two cases. The first case assumes that one of the centers in the query is close to $\mathcal{P}'$. In this case, the sets in $\mathcal{P}'$ are not affecting the cost at all. In the other case, we assume that all of the centers are far from $\mathcal{P}'$ and then we try to show that they are much closer to each other than to the centers relative to the cost of the centers.

Proof of Lemma 3.3. Let $C_w \subseteq \mathcal{X} \times [0, \infty) \times [m]$ be a set of $|C_w| = k$ weighted colored centers. Consider the variables $\tau, \mathcal{P}_1^m, \ldots, \mathcal{P}_k^m$ and $b_1^1, \ldots, b_k^m$ that are computed during the execution of CS-DENSE$(\mathcal{P}, k)$. For every $i \in [m]$ and $r \in [k]$, identify $b_r^i = (x_r^i, j)$. Without the loss of generality, assume that $j = 1$. Therefore, for every $P \in \mathcal{P}$, $i \in [m]$ and $r \in [k]$, we have

$$\tilde{D}(\overline{\text{proj}}(P, (b_r^1, \ldots, b_r^{i-1})), b_r^i) = \tilde{D}\left(P, \left(x_r^i, i\right)\right).$$

For the rest of the proof, let $\bar{b}_r^i = (x_r^i, i)$.

Let $\mathcal{P}_0^0 = \ldots, \mathcal{P}_0^m := \mathcal{P}$. We say that an ordered-set $P \in \mathcal{P}$ is *served* by a colored weighted center $(c, w, t) \in C_w$ if $\tilde{D}(P, C_w) = \tilde{D}(P, (c, w, t))$. For every $i \in [k+1]$, let $(c_i, w_i, t_i) \in C$ denote a center that serves at least $|\mathcal{P}_{i-1}^m|/k$ sets from $\mathcal{P}_{i-1}^m$, and let $\mathcal{P}_i$ denote the sets of $\mathcal{P}$ that are served by (c_i, w_i, t_i). For every $r \in [k]$ and $\ell \in [m]$, let

$$\mathcal{Q}_{r,\ell} \in \underset{\mathcal{Q} \subseteq \mathcal{P}_r^{\ell-1}, |\mathcal{Q}| = \frac{(1-\tau)\cdot\left|\mathcal{P}_r^{\ell-1}\right|}{k}}{\arg\min} \sum_{Q \in \mathcal{Q}} \tilde{D}(Q, \bar{b}_r^\ell), \tag{15}$$

and denote $\tilde{D}_{r,\ell}^* = \sum_{Q \in \mathcal{Q}_{r,\ell}} \tilde{D}(Q, \bar{b}_r^\ell)$.

Since for every $i \in [k]$ we have $|\mathcal{P}_i \cap \mathcal{P}_{i-1}^m| \geq |\mathcal{P}_{i-1}^m|/k$, by the definition of the robust median

$$\sum_{Q \in \mathcal{P}_i \cap \mathcal{P}_{i-1}^m} \tilde{D}(Q, c_i) \geq \tilde{D}^* \left(\mathcal{P}_{i-1}^m, \frac{1}{k}\right). \tag{16}$$

Let $P = (p_1, \ldots, p_m) \in \mathcal{P}$, such that $\tilde{D}(P, C) > 0$. The rest of the proof uses the following case analysis, ***(i)*** there is an index $i \in [k]$ such that

$$\tilde{D}(P, c_i) \leq \frac{16\phi\rho\alpha\tilde{D}_{i,t_i}^*}{|\mathcal{Q}_{k,m}|}, \tag{17}$$

where ϕ and ρ are constants defined in Lemma 2.2 , and ***(ii)*** Otherwise.

Proof for Case *(i)*: By (17),

$$\begin{aligned}
\frac{\tilde{D}(P, C)}{\sum_{Q \in \mathcal{P}} \tilde{D}(Q, C)} &\leq \frac{w_i \tilde{D}(p_{t_i}, c_i)}{\sum_{Q \in \mathcal{P}_i} \tilde{D}(Q, C)} \quad (18)\\
&= \frac{\tilde{D}(p_{t_i}, c_i)}{\sum_{(q_1, \ldots, q_m) \in \mathcal{P}_i} \tilde{D}(q_{t_i}, c_i)} \quad (19)\\
&\leq \frac{16\phi\rho\alpha\tilde{D}_{i,t_i}^*/|\mathcal{Q}_{k,m}|}{\sum_{(q_1, \ldots, q_m) \in \mathcal{P}_i \cap \mathcal{P}_{i-1}^m} \tilde{D}(q_{t_i}, c_i)} \quad (20)\\
&\leq \frac{16\phi\rho\alpha\tilde{D}_{i,t_i}^*/|\mathcal{Q}_{k,m}|}{\tilde{D}_{i,t_i}^*/\alpha} \quad (21)\\
&\leq \frac{16\phi\rho\alpha^2}{|\mathcal{P}_k^m|}, \quad (22)
\end{aligned}$$

where (18) holds since $\mathcal{P}_i$ is a subset of $\mathcal{P}$, (20) holds by the assumption of Case (i), (21) holds by combining (16) and Definition 2.5 (i), and (22) holds since $|\mathcal{Q}_{k,m}| \subseteq |\mathcal{P}_k^m|$.

Proof for Case *(ii)*: By the pigeonhole principle, $c_i = c_j$ for some $i, j \in [k+1]$, such that $i < j$. Put $Q = (q_1, \ldots, q_m) \in \mathcal{P}_j \cap \mathcal{P}_{j-1}^m$ and note that $P \in \mathcal{P}_k^m \subseteq \mathcal{P}_{j-1}^m$. By Markov's inequality, for every $\ell \in [m]$ we have

$$\tilde{D}(P, \bar{b}_{j-1}^\ell), \tilde{D}(Q, \bar{b}_{j-1}^\ell) \leq \frac{2\tilde{D}_{j-1,\ell}^*}{|\mathcal{Q}_{j-1,\ell}|}. \tag{23}$$

By (23) and Lemma 2.2 (i), for every $\ell \in [m]$

$$\tilde{D}(P,Q) \leq \rho\left(\tilde{D}(p_\ell, \bar{b}^\ell_{j-1}) + \tilde{D}(q_\ell, \bar{b}^\ell_{j-1})\right) \leq \frac{4\rho\tilde{D}^*_{j-1,\ell}}{|\mathcal{Q}_{j-1,\ell}|}. \tag{24}$$

Combining the last inequality with Definition 2.1(ii), yields

$$\begin{aligned}
\tilde{D}(P,(c_j,t_j)) - \tilde{D}(Q,(c_j,t_j)) &= \tilde{D}(p_{t_i}, c_j) - \tilde{D}(q_{t_i}, c_j) \\
&\leq \phi\tilde{D}(p_{t_i}, q_{t_i}) + \frac{\tilde{D}(p_{t_i}, c_j)}{4} \qquad (25)\\
&\leq \frac{4\phi\rho\tilde{D}^*_{j-1,t_j}}{|\mathcal{Q}_{j-1,t_j}|} + \frac{\tilde{D}(p_{t_i}, c_j)}{4} \qquad (26)\\
&\leq \frac{4\phi\rho\tilde{D}^*_{j-1,t_j}}{|\mathcal{Q}_{k,m}|} + \frac{\tilde{D}(p_{t_i}, c_j)}{4}, \qquad (27)
\end{aligned}$$

where (25) is followed from Lemma 2.2(ii) and (i) respectively. Finally (26) is obtained after plugging (24) in (25), and (27) is since $|\mathcal{Q}_{k,m}| \leq |\mathcal{Q}_{i,j}|$ for every $i \in [k]$ and $j \in [m]$.

By the assumption of Case(ii), for every $r \in [m]$,

$$\tilde{D}(P,c_j) = \tilde{D}(P,c_i) > \frac{16\phi\rho\alpha\tilde{D}^*_{j,r}}{|\mathcal{Q}_{k,m}|}.$$

Hence

$$\frac{\tilde{D}(P,c_j)}{4} > \frac{4\phi\rho\alpha\tilde{D}^*_{j,r}}{|\mathcal{Q}_{k,m}|}.$$

Combining with (27) yields

$$\tilde{D}(P,c_j) - \tilde{D}(Q,c_j) \leq \frac{\tilde{D}(P,c_j)}{4} + \frac{\tilde{D}(P,c_j)}{4} = \frac{\tilde{D}(P,c_j)}{2}, \tag{28}$$

that is $\tilde{D}(Q,c_j) \geq \tilde{D}(P,c_j)/2$. Hence,

$$\begin{aligned}
\frac{\tilde{D}(P,C)}{\sum_{P'\in\mathcal{P}}\tilde{D}(P',C)} &\leq \frac{\tilde{D}(P,c_j)}{\sum_{Q\in\mathcal{P}_j\cap\mathcal{P}^m_{j-1}}\tilde{D}(Q,c_j)} \\
&\leq \frac{2\tilde{D}(P,c_j)}{\tilde{D}(P,c_j)|\mathcal{P}_j \cap \mathcal{P}^m_{j-1}|} \\
&\leq \frac{2k}{|\mathcal{P}^m_{j-1}|} \qquad (29)\\
&\leq \frac{2k}{|\mathcal{P}^m_k|},
\end{aligned}$$

where (29) holds since $\mathcal{P}_j$ serves at least $\left|\mathcal{P}^m_{j-1}\right|/k$ sets of $\mathcal{P}^m_{j-1}$. $\square$

Corollary C.1. *Let $\mathcal{P}$, k and $(\mathcal{P}^m_k, \mathcal{B}^m_k)$ be as in Lemma 3.3, and let lip, r be as defined in Definition 2.1. Then, for every set $P \in \mathcal{P}^m_k$ and a set $C_w \subseteq \mathcal{X} \times [0,\infty) \times [m]$ of $|C| = k$ weighted colored centers such that $\tilde{D}(P,C) > 0$, we have*

$$\frac{\mathrm{lip}(\tilde{D}(P,C))}{\sum_{P'\in\mathcal{P}}\mathrm{lip}(\tilde{D}(P',C))} \leq \frac{2^r k}{|\mathcal{P}^m_k|}. \tag{30}$$

Proof. In what follows, we use the variables and notations from proof of Lemma 3.3. The proof is is similar to the proof of Lemma 3.3, and is via the following case analysis. ***(i)*** there is an index $i \in [k]$ such that:

$$\tilde{D}(P,c_i) \leq \frac{16\phi\rho\alpha\tilde{D}^*_{i,t_i}}{|\mathcal{Q}_{k,m}|}, \tag{31}$$

and ***(ii)*** Otherwise.

Case *(i)*: By (31),

$$\frac{\operatorname{lip}(\tilde{D}(P,C))}{\sum_{Q\in\mathcal{P}}\operatorname{lip}(\tilde{D}(Q,C))} \le \frac{16\phi\rho\alpha^2}{|\mathcal{P}_k^m|}, \tag{32}$$

similar to (22).

Variation over Case *(ii)*: By (28) we have,

$$2\tilde{D}(Q,c_j) \ge \tilde{D}(P,c_j),$$

and by Definition 2.1,

$$2^r \operatorname{lip}\left(\tilde{D}(Q,c_j)\right) \ge \operatorname{lip}\left(\tilde{D}(P,c_j)\right). \tag{33}$$

Hence,

$$\begin{aligned}
\frac{\operatorname{lip}\left(\tilde{D}(P,C)\right)}{\sum_{P'\in\mathcal{P}}\operatorname{lip}\left(\tilde{D}(P',C)\right)} &\le \frac{\operatorname{lip}\left(\tilde{D}(P,C)\right)}{\sum_{Q\in\mathcal{P}_j\cap\mathcal{P}_{j-1}^m}\operatorname{lip}\left(\tilde{D}(Q,C)\right)}\\
&\le \frac{w_i^r\operatorname{lip}\left(\tilde{D}(P,C)\right)}{w_i^r\sum_{Q\in\mathcal{P}_j\cap\mathcal{P}_{j-1}^m}\operatorname{lip}\left(\tilde{D}(Q,C)\right)}\\
&\le \frac{2^r\operatorname{lip}\left(\tilde{D}(P,c_j)\right)}{\operatorname{lip}\left(\tilde{D}(P,c_j)\right)|\mathcal{P}_j\cap\mathcal{P}_{j-1}^m|}\\
&\le \frac{2^r k}{|\mathcal{P}_{j-1}^m|} \qquad (34)\\
&\le \frac{2^r k}{|\mathcal{P}_k^m|},
\end{aligned}$$

where (34) holds since $\mathcal{P}_j$ serves at least $\left|\mathcal{P}_{j-1}^m\right|/k$ sets of $\mathcal{P}_{j-1}^m$. $\square$

The following claim has nothing to do with the correctness of Algorithm 1. However, it will be used later in the proof of Lemma 3.4.

Claim C.2. *Let $\mathcal{P}$ be an (n,m)-ordered-set in $(\mathcal{X},D)$, and $k\ge 1$ be an integer. For every integer $i\in[m]$, and ordered set $P\in\mathcal{P}$, we have*

$$\sup_{\substack{C_w\subseteq\mathcal{X}\times\mathbb{R}\times[m]\\|C_w|=k,\ \tilde{D}(P,C_w)>0}} \frac{\tilde{D}(P,C_w)}{\sum_{Q\in\mathcal{P}}\tilde{D}(Q,C_w)} \ge \sup_{\substack{C_w\subseteq\mathcal{X}\times\mathbb{R}\times[i]\\|C_w|=k,\ \tilde{D}(P,C_w)>0}} \frac{\tilde{D}((p_1,\ldots,p_i),C_w)}{\sum_{(q_1,\ldots,q_m)\in\mathcal{P}}\tilde{D}((q_1,\ldots,q_i),C_w)}, \tag{35}$$

i.e., the sensitivity of any prefix with respect to all the other sets prefixes is smaller than the sensitivity of the original set with respect to the original family.

Proof. Let $i\in[m]$ be an integer. Let $C\subseteq\mathcal{X}\times(0,\infty)\times[i]$ be a set that maximizes the right hand side of (35), i.e.,

$$\sup_{\substack{C_w\subseteq\mathcal{X}\times\mathbb{R}\times[i]\\|C_w|=k,\ \tilde{D}(P,C_w)>0}} \frac{\tilde{D}((p_1,\ldots,p_i),C_w)}{\sum_{(q_1,\ldots,q_m)\in\mathcal{P}}\tilde{D}((q_1,\ldots,q_i),C_w)} = \frac{\tilde{D}((p_1,\ldots,p_i),C)}{\sum_{(q_1,\ldots,q_m)\in\mathcal{P}}\tilde{D}((q_1,\ldots,q_i),C)}.$$

Then

$$\sup_{\substack{C_w\subseteq\mathcal{X}\times\mathbb{R}\times[m]\\|C_w|=k,\ \tilde{D}(P,C_w)>0}} \frac{\tilde{D}(P,C_w)}{\sum_{Q\in\mathcal{P}}\tilde{D}(Q,C_w)} \ge \frac{\tilde{D}(P,C)}{\sum_{Q\in\mathcal{P}}\tilde{D}(Q,C)} = \tag{36}$$

$$\sup_{\substack{C_w\subseteq\mathcal{X}\times\mathbb{R}\times[i]\\|C_w|=k,\ \tilde{D}(P,C_w)>0}} \frac{\tilde{D}((p_1,\ldots,p_i),C_w)}{\sum_{(q_1,\ldots,q_m)\in\mathcal{P}}\tilde{D}((q_1,\ldots,q_i),C_w)} \tag{37}$$

where (36) holds since $\mathcal{X}\times(0,\infty)\times[i]\subseteq\mathcal{X}\times(0,\infty)\times[m]$, and (37) is by the definition of C. $\square$

C.2 Correctness of Algorithm 2

Claim C.3. *Let p be a point on a line ℓ in $\mathbb{R}^d$, and let $\mathcal{S} = \{c \in \mathbb{R}^d | \, \|p - c\| = 1\}$ denote the unit sphere that is centered at p; see Fig. 5. Then, for every point $q \in \mathcal{S}$, we have*

$$\operatorname{dist}(\ell, q) \leq \sqrt{2} \cdot \operatorname{dist}(\mathcal{S} \cap \ell, q).$$

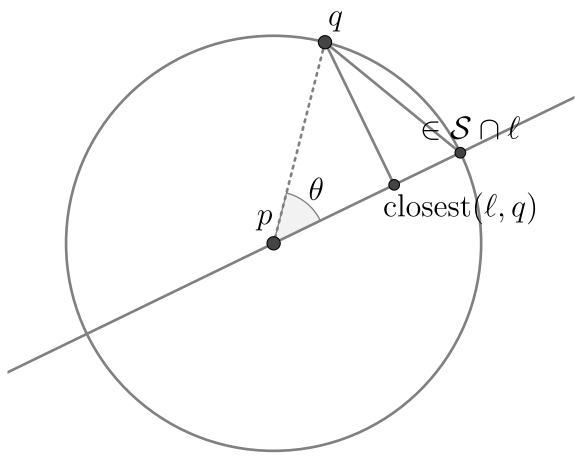


Figure 5

Proof of Claim C.3. Let a, b be a pair of unit vectores in $\mathbb{R}^d$ such that $a^T b \geq 0$.

$$\|a - b\| = \sqrt{\|a - b\|^2} = \sqrt{2\,\|1 - a^T b\|} \tag{38}$$

$$\leq \sqrt{2\,\|1 - (a^T b)^2\|} = \sqrt{2\operatorname{dist}^2(a, \operatorname{sp}(b))} = \sqrt{2}\operatorname{dist}(a, \operatorname{sp}(b)) \tag{39}$$

Then if p were on the origin by substitute $a = q$ and $b = \mathcal{S} \cap \ell$ we get

$$\operatorname{dist}(\ell, q) \leq \sqrt{2}\operatorname{dist}(\mathcal{S} \cap \ell, q).$$

□

Lemma C.4. *Let $\mathcal{L}$ be an (n, m)-set of lines, and $B = (b_1, \ldots, b_m)$ be a set of m points, both in $\mathbb{R}^d$, such that for every set $L = (\ell_1, \ldots, \ell_m) \in \mathcal{L}$ and every $i \in [m]$, the line ℓ_i intersects b_i. Let $k \geq 1$ be an integer, and let $s : \mathcal{L} \to (0, 1]$ be the output of call to* GROUPED-SENSITIVITY$(\mathcal{L}, B, k)$*;see Algorithm 2. Then, for every $L \in \mathcal{L}$, we have*

$$s(L) \geq S_{\mathcal{L},k}(L).$$

Proof of Lemma C.4. Define $P(L)$ for every $L \in \mathcal{L}$, as in Algorithm 2. For every line $\ell \subseteq \mathbb{R}^d$, $p' \in \ell$, and $p \in \mathbb{R}^d \setminus \{p'\}$, by Thales Theorem, we have

$$\operatorname{dist}(\ell, p) = \operatorname{dist}(p, p') \cdot \operatorname{dist}\left(\ell, p' + \frac{p - p'}{\operatorname{dist}(p, p')}\right). \tag{40}$$

Let $L = (\ell_1, \ldots, \ell_m) \in \mathcal{L}$, and $C \subseteq \mathbb{R}^d$ be a set of $|C| = k$ centers such that $\tilde{D}(L, C) > 0$, and recall that, by (4), for every set P in $\mathcal{X}$ let $\operatorname{closest}(C, P)$ denote the only point in $\operatorname{closest}(C, P, \frac{1}{|C|})$ consists of the closest point in C to a set in P. Hence,

$$\begin{aligned}
\tilde{D}(L, C) &= \min_{i \in [m]} \tilde{D}(\ell_i, C) \\
&= \min_{i \in [m]} \operatorname{lip}\left(\operatorname{dist}(\ell_i, \operatorname{closest}(C, \ell_i))\right) \\
&= \min_{i \in [m]} \operatorname{lip}\left(\operatorname{dist}(\operatorname{closest}(C, \ell_i), b_i) \cdot \operatorname{dist}\left(\ell_i, b_i + \frac{\operatorname{closest}(C, \ell_i) - b_i}{\operatorname{dist}\left(\operatorname{closest}(C, \ell_i), b_i\right)}\right)\right), \quad (41)
\end{aligned}$$

where (41) holds by (40).

For every $i \in [m]$, and every point $q \in \mathbb{R}^d \setminus \{b_i\}$, the point $b_i + \frac{q-b_i}{\|q-b_i\|}$ is in $\mathbb{S}_i$, i.e., on the unit sphere that is centered at b_i. By Claim C.3,

$$\frac{\tilde{D}(L,C)}{\sum_{L'\in\mathcal{L}} \tilde{D}(L',C)} \leq \frac{\sqrt{2^r}\cdot \min_{i\in[m]} \operatorname{lip}\left(\operatorname{dist}(\operatorname{closest}(C,\ell_i),b_i)\cdot \operatorname{dist}\left(\ell_i\cap \mathbb{S}_i, b_i + \frac{\operatorname{closest}(C,\ell_i)-b_i}{\|\operatorname{closest}(C,\ell_i)-b_i\|}\right)\right)}{\sum\limits_{(\ell_1',\ldots,\ell_m')\in\mathcal{L}} \min_{j\in[m]} \operatorname{lip}\left(\operatorname{dist}(\operatorname{closest}(C,\ell_j'),b_j)\cdot \operatorname{dist}\left(\ell_j'\cap \mathbb{S}_j, b_j + \frac{\operatorname{closest}(C,\ell_j')-b_j}{\left\|\operatorname{closest}(C,\ell_j')-b_j\right\|}\right)\right)}. \tag{42}$$

Let

$$C' = \left\{ \left(b_i + \frac{c-b_i}{\|c-b_i\|}, \|c-b\|, i\right) \,\middle|\, i\in[m], c\in C\right\}. \tag{43}$$

Hence, we can reformulate the right hand side of (42) to

$$\frac{\tilde{D}(L,C)}{\sum_{L'\in\mathcal{L}} \tilde{D}(L',C)} \leq \frac{\sqrt{2^r}\cdot \min_{(c,w,t)\in C'} \operatorname{lip}\left(w\cdot \operatorname{dist}\left(\ell_t\cap\mathbb{S}_t, c\right)\right)}{\sum_{(\ell_1',\ldots,\ell_m')\in\mathcal{L}} \min_{(c',w',t')\in C'} \operatorname{lip}\left(w'\cdot \operatorname{dist}\left(\ell_{t'}\cap\mathbb{S}_{t'}, c'\right)\right)}. \tag{44}$$

Since $C' \subseteq \mathbb{R}^d \times (0,\infty)\times[m]$, and the cardinality of the set C' is at most $|C|\cdot|B| = mk$, we have

$$S_{\mathcal{L},k}(L) = \sup_{\substack{C\subseteq\mathbb{R}^d, |C|=k\\ \tilde{D}(L,C)>0}} \frac{\tilde{D}(L,C)}{\sum_{L'\in\mathcal{L}} \tilde{D}(L',C)} \tag{45}$$

$$\leq \sup_{\substack{C_w\subseteq\mathbb{R}^d\times(0,\infty)\times[m]\\ |C_w|=mk}} \frac{\sqrt{2^r}\cdot \min_{(c,w,t)\in C_w} \operatorname{lip}\Big(w\cdot \operatorname{dist}\left(\ell_t\cap\mathbb{S}_t, c\right)\Big)}{\sum_{\{\ell_1',\ldots,\ell_m'\}\in\mathcal{L}} \min_{(c',w',t')\in C_w} \operatorname{lip}\Big(w'\cdot \operatorname{dist}\left(\ell_{t'}\cap\mathbb{S}_{t'}, c'\right)\Big)} \tag{46}$$

$$\geq \sup_{\substack{C_w\subseteq\mathbb{R}^d\times(0,\infty)\times[2m]\\ |C_w|=2mk}} \frac{\sqrt{2^r}\operatorname{lip}\left(\tilde{D}\left(P(L),C_w\right)\right)}{\sum_{L'\in\mathcal{L}} \operatorname{lip}\left(\tilde{D}\left(P(L'),C_w\right)\right)} \leq \sqrt{2^r}\cdot s'(P(L)), \tag{47}$$

where (45) is by Definition 3.1, (46) is by (44), the left hand side of (47) is by the definition of $\tilde{D}$, and the right hand side of (47) is by Corollary C.1. Also the two factor in the size of C_w is since each line represented by to points with different colors (we may avoid this by modifying the nations in Section 3.1 but we leave the proof for future version). Finally,

$$s(L) = \sqrt{2^r}\cdot s'(P(L)) \geq S_{\mathcal{L},k}(L),$$

where the left hand side is by Algorithm 2, and the right hand side is by (47). □

C.3 Correctness of Algorithm 3

Lemma C.5 (restatement of Lemma 3.4)**.** *Let $\mathcal{L}$ be an (n,m)-set of lines in $\mathbb{R}^d$, and $k \geq 1$ be an integer. Let $\left(\mathcal{L}^{m+1}, \mathcal{B}^m\right)$ be the output of a call to* LS-DENSE$(\mathcal{L},k)$*; see Algorithm 3. Then, for every $L \in \mathcal{L}^{m+1}$, we have $S_{\mathcal{L},k}(L) \in O(k) \cdot \left(\frac{1}{|\mathcal{L}^{m+1}|}\right)$.*

The proof of this lemma is inspired by the proof of Lemma 4.1 in [26]. The proof uses the following pair of lemmas. Lemma B.1 [26] whose assumptions hold also for sets of lines, and a generalization of Lemma B.2 [26] for parallel lines.

Lemma C.6 (Lemma B.1 in [26])**.** *Let $k \geq 1$ be an integer. Let A, B be a pair of sets of lines in $\mathbb{R}^d$, and $C \subseteq \mathbb{R}^d$ be a set of $|C| = k$ points. If $\tilde{D}(A \cup B, C) \neq \tilde{D}(B,C)$ then $\tilde{D}(A \cup B, C) = \tilde{D}(A,C)$.*

Lemma C.7 (Generalization of Lemma B.2 in [26])**.** *Let A be a finite set of lines in $\mathbb{R}^d$, let $\ell \in A$ and ℓ' be a line in $\mathbb{R}^d$ that is parallel to ℓ. Let $B = (A \setminus \{\ell\}) \cup \{\ell'\}$. Then, for every $C \subseteq \mathbb{R}^d$, we have*

$$\tilde{D}(A,C) \leq \rho\left(\tilde{D}(B,C) + \tilde{D}(\ell,\ell')\right).$$

Proof. By definition, we have

$$\begin{aligned}
\tilde{D}(A,C) &= \min\left\{\tilde{D}(\ell,C), \tilde{D}(A \setminus \{\ell\}, C)\right\} \\
&\leq \min\left\{\rho\left(\tilde{D}(\ell,\ell') + \tilde{D}(\ell',C)\right), \tilde{D}(A \setminus \{\ell\}, C)\right\} && (48)\\
&\leq \min\left\{\rho\left(\tilde{D}(\ell,\ell') + \tilde{D}(\ell',C)\right), \rho\left(\tilde{D}(A \setminus \{\ell\}, C) + \tilde{D}(\ell,\ell')\right)\right\} \\
&\leq \rho \min\left\{\tilde{D}(\ell',C), \tilde{D}(A \setminus \{\ell\}, C)\right\} + \rho\tilde{D}(\ell,\ell') \\
&= \rho(\tilde{D}(B,C) + \tilde{D}(\ell,\ell')), && (49)
\end{aligned}$$

where (48) holds since the distance from a line to a parallel line is the same from every point on the line, hence the weak triangle holds by Definition 2.1, and (49) is by the definition of B. □

Proof of Lemma 3.4. In what follows, we use the variables and notations from Algorithm 3. Put $L \in \mathcal{L}^{m+1}$, $i \in [m]$, and consider the ith iteration of the "for" loop at Line 3 of Algorithm 3. Let $C \subseteq \mathbb{R}^d$ be a set of $|C| = k$ centers such that $\tilde{D}(L,C) > 0$ and $\tilde{D}(T(L, \mathcal{B}^m), C) > 0$. Let

$$\widehat{\mathcal{L}}^{i-1} := \left\{Q \in \mathcal{L}^{i-1} \middle| \tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C) = \tilde{D}(\mathrm{proj}(Q,\mathcal{B}^{i-1}),C)\right\}$$

be the union of sets $Q \in \mathcal{L}^{i-1}$ whose closest projected line on $\mathcal{B}^{i-1}$ to the query C is among the lines that are translated to the points of $\mathcal{B}^{i-1}$. Firstly, we prove that

$$\frac{\tilde{D}(\mathrm{T}(L,\mathcal{B}^{i-1}),C)}{\sum\limits_{Q \in \mathcal{L}^{i-1}} \tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C)} \leq 5\rho^2 \frac{\tilde{D}(\mathrm{T}(L,\mathcal{B}^{i}),C)}{\sum\limits_{Q \in \mathcal{L}^{i}} \tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i}),C)} + \frac{4\rho}{|\mathcal{L}^i|} \tag{50}$$

via the following case analysis: **(i)** $\left|\widehat{\mathcal{L}}^{i-1}\right| \geq \frac{|\mathcal{L}^{i-1}|}{2}$, i.e., more than half of the sets satisfy that their closest line to C is amongst their translated lines onto $\mathcal{B}^{i-1}$, and **(ii)** Otherwise, i.e., $\left|\widehat{\mathcal{L}}^{i-1}\right| < \frac{|\mathcal{L}^{i-1}|}{2}$.

Proof for Case (i): $\left|\widehat{\mathcal{L}}^{i-1}\right| \geq \frac{|\mathcal{L}^{i-1}|}{2}$. By Line 5 in Algorithm 3, we have

$$\mathcal{L}^i \subseteq \mathcal{L}^{i-1} \subseteq \cdots \subseteq \mathcal{L}^0 = \mathcal{L}. \tag{51}$$

Therefore,

$$\begin{aligned}
\sum_{Q \in \mathcal{L}^{i-1}} \tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C) &\geq \sum_{Q \in \widehat{\mathcal{L}}^{i-1}} \tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C) && (52)\\
&= \sum_{Q \in \widehat{\mathcal{L}}^{i-1}} \tilde{D}(\mathrm{proj}(Q,\mathcal{B}^{i-1}),C), && (53)
\end{aligned}$$

where (52) holds since $\widehat{\mathcal{L}}^{i-1} \subseteq \mathcal{L}^{i-1}$, and (53) is by the definition of $\widehat{\mathcal{L}}^{i-1}$. This proves (50) for Case (i) as

$$\begin{aligned}\frac{\tilde{D}(\mathrm{T}(L,\mathcal{B}^{i-1}),C)}{\sum\limits_{Q\in\mathcal{L}^{i-1}}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C)} &\leq \frac{\tilde{D}(\mathrm{proj}(L,\mathcal{B}^{i-1}),C)}{\sum\limits_{Q\in\mathcal{L}^{i-1}}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C)}\\ &\leq \frac{\tilde{D}(\mathrm{proj}(L,\mathcal{B}^{i-1}),C)}{\sum_{Q\in\widehat{\mathcal{L}}^{i-1}}\tilde{D}(\mathrm{proj}(Q,\mathcal{B}^{i-1}),C)} &(54)\\ &\leq \frac{\tilde{D}(\mathrm{proj}(L,\mathcal{B}^{i-1}),C)}{\sum_{Q\in\widehat{\mathcal{L}}^{m}}\tilde{D}(\mathrm{proj}(Q,\mathcal{B}^{m}),C)}, &(55)\end{aligned}$$

where the first inequality holds since $\mathrm{T}(L,\mathcal{B}^{i-1}) \supseteq \mathrm{proj}(L,\mathcal{B}^{i-1})$ by Definition 2.7, the second inequality is by (53), and the third is a simple corollary from combining Claim C.2, and Lemma C.4.

Proof for Case (ii): $\left|\widehat{\mathcal{L}}^{i-1}\right| < \frac{|\mathcal{L}^{i-1}|}{2}$. Let $\gamma = 1/(2k)$. Let b^i, $\mathcal{L}^i$, and $\bar{\mathcal{L}}^{i-1}$ be as defined in Lines 4, and 5, 7, respectively. Identify $\mathcal{B}^{i-1} = \left\{b^1,\ldots,b^{i-1}\right\}$ if $i \geq 2$, and $\mathcal{B}^{i-1} = \emptyset$ if $i = 1$. For every $Q \in \mathcal{L}^{i-1}$, substituting $A = \overline{\mathrm{proj}}(Q,\mathcal{B}^{i-1})$ and $B = \mathrm{proj}(Q,\mathcal{B}^{i-1})$ in Lemma C.6 yields

$$\begin{aligned}&\left\{Q \in \mathcal{L}^{i-1} | \tilde{D}\big(\mathrm{T}(Q,\mathcal{B}^{i-1}),C\big) \neq \tilde{D}(\mathrm{proj}(Q,\mathcal{B}^{i-1}),C)\right\}\\ &\subseteq \left\{Q \in \mathcal{L}^{i-1} | \tilde{D}\big(\mathrm{T}(Q,\mathcal{B}^{i-1}),C\big) = \tilde{D}(\overline{\mathrm{proj}}(Q,\mathcal{B}^{i-1}),C)\right\}.\end{aligned} \tag{56}$$

Hence,

$$\begin{aligned}&\left|\left\{Q \in \mathcal{L}^{i-1} \;\middle|\; \tilde{D}\left(\mathrm{T}(Q,\mathcal{B}^{i-1}),C\right) = \tilde{D}\left(\overline{\mathrm{proj}}(Q,\mathcal{B}^{i-1}),C\right)\right\}\right|\\ \geq &\left|\left\{Q \in \mathcal{L}^{i-1} \;\middle|\; \tilde{D}\left(\mathrm{T}(Q,\mathcal{B}^{i-1}),C\right) \neq \tilde{D}(\mathrm{proj}(Q,\mathcal{B}^{i-1}),C)\right\}\right| &(57)\\ = &\left|\mathcal{L}^{i-1} \setminus \widehat{\mathcal{L}}^{i-1}\right| \geq \frac{\left|\mathcal{L}^{i-1}\right|}{2}, &(58)\end{aligned}$$

where(57) is by (56), the equality in (58) is by the definitions of $\mathcal{L}^{i-1}$ and $\widehat{\mathcal{L}}^{i-1}$, and the last inequality is by the assumption of Case (ii).

Recall that by Line 7 of Algorithm 3,

$$\bar{\mathcal{L}}^{i-1} = \left\{\overline{\mathrm{proj}}(Q,\mathcal{B}^{i-1}) \middle| Q \in \mathcal{L}^{i-1}\right\},$$

and define

$$Z = \left\{Q \in \mathcal{L}^{i-1} \middle| \overline{\mathrm{proj}}(Q,\mathcal{B}^{i-1}) \in \mathrm{closest}\left(\bar{\mathcal{L}}^{i-1},C,\frac{1}{2}\right)\right\}.$$

Since Z contains the $|Z| \leq \left\lceil \frac{|\mathcal{L}^{i-1}|}{2} \right\rceil$ sets $Q \in \mathcal{L}^{i-1}$ with the smallest distance $\tilde{D}(\overline{\mathrm{proj}}(Q,\mathcal{B}^{i-1}),C)$, for any set $Z' \subseteq \mathcal{L}^{i-1}$ such that $|Z'| \geq \frac{|\mathcal{L}^{i-1}|}{2}$, we have

$$\sum_{Q\in Z}\tilde{D}(\overline{\mathrm{proj}}(Q,\mathcal{B}^{i-1}),C) \leq \sum_{Q\in Z'}\tilde{D}(\overline{\mathrm{proj}}(Q,\mathcal{B}^{i-1}),C). \tag{59}$$

By the assumption of Case (ii),

$$\left|\mathcal{L}^{i-1} \setminus \widehat{\mathcal{L}}^{i-1}\right| \geq \frac{\left|\mathcal{L}^{i-1}\right|}{2}, \tag{60}$$

and by the definition of Z, we have

$$\left\{\overline{\mathrm{proj}}(Q,\mathcal{B}^{i-1}) \middle| Q \in Z\right\} = \mathrm{closest}\left(\bar{\mathcal{L}}^{i-1},C,\frac{1}{2}\right). \tag{61}$$

Therefore,

$$\begin{aligned}\sum_{Q\in\mathrm{closest}(\bar{\mathcal{L}}^{i-1},C,1/2)}\tilde{D}(Q,C) &= \sum_{Q\in Z}\tilde{D}(\overline{\mathrm{proj}}(Q,\mathcal{B}^{i-1}),C)\\ &\leq \sum_{Q\in\mathcal{L}^{i-1}\setminus\widehat{\mathcal{L}}^{i-1}}\tilde{D}(\overline{\mathrm{proj}}(Q,\mathcal{B}^{i-1}),C),\end{aligned} \tag{62}$$

where the equality is by (61), and the inequality is by substituting $Z' = \mathcal{L}^{i-1} \setminus \widehat{\mathcal{L}}^{i-1}$ in (59). By the definitions of $\mathcal{L}^{i-1}$ and $\widehat{\mathcal{L}}^{i-1}$, for every $Q \in \mathcal{L}^{i-1} \setminus \widehat{\mathcal{L}}^{i-1}$, we have

$$\tilde{D}(\overline{\mathrm{proj}}(Q, \mathcal{B}^{i-1}), C) = \tilde{D}(\mathrm{T}(Q, \mathcal{B}^{i-1}), C). \tag{63}$$

Let

$$\mathrm{OPT}_i = \min_{C' \subseteq \mathbb{R}^d, |C'|=k} \tilde{D}\left(\mathrm{closest}(\bar{\mathcal{L}}^{i-1}, C', 1/2), C'\right). \tag{64}$$

Hence,

$$\begin{aligned}
\mathrm{OPT}_i &\leq \sum_{\bar{Q} \in \mathrm{closest}(\bar{\mathcal{L}}^{i-1}, C, 1/2)} \tilde{D}(\bar{Q}, C) && (65)\\
&\leq \sum_{Q \in \mathcal{L}^{i-1} \setminus \widehat{\mathcal{L}}^{i-1}} \tilde{D}(\overline{\mathrm{proj}}(Q, \mathcal{B}^{i-1}), C) && (66)\\
&= \sum_{Q \in \mathcal{L}^{i-1} \setminus \widehat{\mathcal{L}}^{i-1}} \tilde{D}(\mathrm{T}(Q, \mathcal{B}^{i-1}), C) && (67)\\
&\leq \sum_{Q \in \mathcal{L}^{i-1}} \tilde{D}(\mathrm{T}(Q, \mathcal{B}^{i-1}), C), && (68)
\end{aligned}$$

where (65) holds by the definition of OPT_i, (66) is by (62), (67) is by (63), and (68) holds since $\mathcal{L}^{i-1} \setminus \widehat{\mathcal{L}}^{i-1} \subseteq \mathcal{L}^{i-1}$.

Recall that $\mathcal{B}^m = (b_1, \ldots, b_m)$ is an ordered set. Denote the closest line to b_1 in L by ℓ_1, i.e., $\ell_1 \in \arg\min_{\ell \in L} \tilde{D}(\ell, b_1)$. For every $j \in [m-1]$, recursively define ℓ_{j+1} to be the line that is closest to b_i over every line in $L \setminus \{\ell_1, \ldots, \ell_j\}$, i.e.,

$$\ell_{j+1} \in \operatorname*{arg\,min}_{\ell \in L \setminus \{\ell_1, \ldots, \ell_i\}} \tilde{D}(\ell, b_{j+1}). \tag{69}$$

Hence, for every $j \in [m]$, we have

$$\tilde{D}\left(\overline{\mathrm{proj}}(L, \mathcal{B}^{j-1}), b_j\right) = \tilde{D}(\ell_j, b_j). \tag{70}$$

Since $L \in \mathcal{L}^{m+1} \subseteq \mathcal{L}^i$ and $\gamma = \frac{1}{2k}$, by Line 5 of Algorithm 3 we have

$$\overline{\mathrm{proj}}(L, \mathcal{B}^{i-1}) \in \mathrm{closest}\left(\bar{\mathcal{L}}^{i-1}, \left\{b^i\right\}, \frac{(1-\tau)\gamma}{2}\right). \tag{71}$$

By the Pigeonhole Principle, the largest cluster in every set C' of k centers contains at least $\frac{|\bar{\mathcal{L}}^{i-1}|}{k} \leq \frac{|\bar{\mathcal{L}}^{i-1}|}{2k} = \gamma|\bar{\mathcal{L}}^{i-1}|$ sets. Since, by Line 4 of Algorithm 3, b_i is a $(\gamma, \tau, 4)$-median, we have

$$\sum_{Q \in \mathrm{closest}\left(\bar{\mathcal{L}}^{i-1}, \{b_i\}, (1-\tau)\gamma\right)} \tilde{D}(Q, b_i) \leq 4 \min_{b \in \mathbb{R}^d} \sum_{Q \in \mathrm{closest}\left(\bar{\mathcal{L}}^{i-1}, \{b\}, \gamma\right)} \tilde{D}\left(Q, b\right) \leq 4 \cdot \mathrm{OPT}_i. \tag{72}$$

Therefore,

$$\begin{aligned}
\tilde{D}(\ell_i, b_i) &= \tilde{D}\left(\overline{\mathrm{proj}}(L, \mathcal{B}^{i-1}), b_i\right) && (73)\\
&\leq 4 \cdot \frac{\sum_{Q \in \mathrm{closest}\left(\bar{\mathcal{L}}^{i-1}, \{b_i\}, (1-\tau)\gamma\right)} \tilde{D}(Q, b_i)}{(1-\tau)\gamma \left|\bar{\mathcal{L}}^{i-1}\right|} && (74)\\
&\leq 4 \cdot \frac{\sum_{Q \in \mathrm{closest}(\bar{\mathcal{L}}^{i-1}, \{b_i\}, (1-\tau)\gamma)} \tilde{D}(Q, b_i)}{|\mathcal{L}^i|} && (75)\\
&\leq \frac{8\mathrm{OPT}_i}{|\mathcal{L}^i|}, && (76)
\end{aligned}$$

where (73) is by (70), (74) is by combining Markov's inequality with (71), (75) follows since $|\mathcal{L}^i| = \left\lceil \frac{(1-\tau)\gamma}{2} |\mathcal{L}^{i-1}| \right\rceil = \left\lceil \frac{(1-\tau)\gamma}{2} |\bar{\mathcal{L}}^{i-1}| \right\rceil \leq (1-\tau)\gamma|\mathcal{L}^{i-1}|$, and (76) is by (72).

Now, since

$$\mathrm{T}(L,\mathcal{B}^i) = \left(\mathrm{T}(L,\mathcal{B}^{i-1}) \setminus \{\ell_i\}\right) \cup \{\mathrm{T}(\ell_i,b_i)\},$$

i.e., the sets $\mathrm{T}(L,\mathcal{B}^{i-1})$ and $\mathrm{T}(L,\mathcal{B}^i)$ differ only one line, and by Definition 2.7 the line $\mathrm{T}(\ell_i,b_i)$ is parallel to ℓ_i. Thus, by substituting $A = \mathrm{T}(L,\mathcal{B}^{i-1})$, $B = \mathrm{T}(L,\mathcal{B}^i)$, and $\ell = \mathrm{T}(\ell_i,b_i)$ in Lemma C.7, we obtain

$$\tilde{D}(\mathrm{T}(L,\mathcal{B}^{i-1}),C) \leq \rho\tilde{D}(\mathrm{T}(P,\mathcal{B}^i),C) + \rho\tilde{D}(\ell_i,\mathrm{T}(\ell_i,b_i)). \tag{77}$$

Dividing both sides of (77) by $\sum_{Q\in\mathcal{L}^{i-1}} \tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C)$ yields

$$\frac{\tilde{D}(\mathrm{T}(L,\mathcal{B}^{i-1}),C)}{\sum_{Q\in\mathcal{L}^{i-1}}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C)} \leq \frac{\rho\tilde{D}(\mathrm{T}(L,\mathcal{B}^{i}),C)}{\sum_{Q\in\mathcal{L}^{i-1}}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C)} + \frac{\rho\tilde{D}(\ell_i,\mathrm{T}(\ell_i,b_i))}{\sum_{Q\in\mathcal{L}^{i-1}}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C)}. \tag{78}$$

The rightmost term in (78) can then be bounded by

$$\begin{align}
\frac{\rho\tilde{D}\left(\ell_i,\mathrm{T}(\ell_i,b_i)\right)}{\sum_{Q\in\mathcal{L}^{i-1}}\tilde{D}\left(\mathrm{T}(Q,\mathcal{B}^{i-1}),C\right)} &\leq \frac{\rho\tilde{D}(\ell_i,\mathrm{T}(\ell_i,b_i))}{\mathrm{OPT}_i} \tag{79}\\
&= \frac{\rho\tilde{D}\left(\ell_i,b_i\right)}{\mathrm{OPT}_i} \tag{80}\\
&\leq \frac{8\rho\mathrm{OPT}_i}{|\mathcal{L}^i|\,\mathrm{OPT}_i} = \frac{8\rho}{|\mathcal{L}^i|}, \tag{81}
\end{align}$$

where (79) is by (68), and the inequality in (81) holds by (76).

We now bound the middle term of (78). Similarly to (69), for every $Q \in \mathcal{L}^i$ identify $Q = \{q_1,\ldots,q_m\}$. We have,

$$\begin{align}
\sum_{Q\in\mathcal{L}^i}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^i),C) &\leq \rho\sum_{Q\in\mathcal{L}^i}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C) + \rho\sum_{Q\in\mathcal{L}^i}\tilde{D}(q_i,\mathrm{T}(q_i,b_i)) \tag{82}\\
&\leq \rho\sum_{Q\in\mathcal{L}^i}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C) + \rho\left|\mathcal{L}^i\right|\frac{4\mathrm{OPT}_i}{|\mathcal{L}^i|} \tag{83}\\
&\leq \rho\sum_{Q\in\mathcal{L}^{i-1}}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C) + 4\rho\mathrm{OPT}_i \tag{84}\\
&\leq (5\rho)\sum_{Q\in\mathcal{L}^{i-1}}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C), \tag{85}
\end{align}$$

where (82) holds by summing (77) over every $Q \in \mathcal{L}^i$, (83) holds since b_i is robst midian for $\bar{\mathcal{L}}^{i-1}$, (84) holds since $\mathcal{L}^i \subseteq \mathcal{L}^{i-1}$ by (51), and (85) is by (68). By (85), the middle term of (78) is bounded by

$$\frac{\rho\tilde{D}(\mathrm{T}(L,\mathcal{B}^i),C)}{\sum_{Q\in\mathcal{L}^{i-1}}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C)} \leq \frac{5\rho^2\tilde{D}(\mathrm{T}(L,\mathcal{B}^i),C)}{\sum_{Q\in\mathcal{L}^{i}}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i}),C)}. \tag{86}$$

Combining (78), (81) and (86) yields (50) as

$$\frac{\tilde{D}(\mathrm{T}(L,\mathcal{B}^{i-1}),C)}{\sum_{Q\in\mathcal{L}^{i-1}}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i-1}),C)} \leq \frac{5\rho^2\tilde{D}(\mathrm{T}(L,\mathcal{B}^i),C)}{\sum_{Q\in\mathcal{L}^{i}}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^{i}),C)} + \frac{8\rho}{|\mathcal{L}^i|}.$$

Wrapping all together. We can now apply (50) recursively over every $i \in [m]$ to obtain

$$\begin{align}
\frac{\tilde{D}(L,C)}{\sum_{Q\in\mathcal{L}}\tilde{D}(Q,C)} &= \frac{\tilde{D}(\mathrm{T}(L,\mathcal{B}^0),C)}{\sum_{Q\in\mathcal{L}^0}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^0),C)} \nonumber\\
&\leq (5\rho^2)^m\frac{\tilde{D}(\mathrm{T}(L,\mathcal{B}^m),C)}{\sum_{Q\in\mathcal{L}^m}\tilde{D}(\mathrm{T}(Q,\mathcal{B}^m),C)} + 4\rho\sum_{i=1}^{m}\frac{(5\rho^2)^{i-1}}{|\mathcal{L}^i|}. \tag{87}
\end{align}$$

Furthermore, for every $L' \in \mathcal{L}^{m+1}$ by Lines 9, and 10 of Algorithm 3 and by combining Lemma 3.3 and Lemma C.4 we get

$$\frac{\tilde{D}(\mathrm{T}(L', \mathcal{B}^m), C)}{\sum_{Q \in \mathcal{L}^m} \tilde{D}(\mathrm{T}(Q, \mathcal{B}^m), C)} \leq \frac{2\sqrt{2}mk}{|\mathcal{L}^{m+1}|}. \tag{88}$$

Lemma 3.4 now holds as

$$S_{\mathcal{L},k} = \frac{\tilde{D}(L, C)}{\sum_{Q \in \mathcal{L}} \tilde{D}(Q, C)} \leq \frac{2\sqrt{2}mk(5\rho^2)^m}{|\mathcal{L}^{m+1}|} + 4\rho \sum_{i=1}^{m} \frac{(5\rho^2)^{i-1}}{|\mathcal{L}^i|} \tag{89}$$

$$\leq \frac{2\sqrt{2}mk(5\rho^2)^m}{|\mathcal{L}^{m+1}|} + 4\rho \sum_{i=1}^{m} \frac{(5\rho^2)^{i-1}}{|\mathcal{L}^{m+1}|} \tag{90}$$

$$\leq \frac{2\sqrt{2}mk(5\rho^2)^m}{|\mathcal{L}^{m+1}|} + \frac{4\rho}{|\mathcal{L}^{m+1}|} \cdot \frac{(5\rho^2)^{m-1} - 1}{(5\rho^2) - 1} \tag{91}$$

$$\leq \frac{2\sqrt{2}mk(5\rho^2)^m}{|\mathcal{L}^{m+1}|} + \frac{4\rho}{|\mathcal{L}^{m+1}|} \cdot (5\rho^2)^m \tag{92}$$

$$\leq \frac{15mk\rho(5\rho^2)^m}{|\mathcal{L}^{m+1}|}, \tag{93}$$

where (89) holds by plugging (88) in (87), (90) holds since $\left|\mathcal{L}^{m+1}\right| \leq \left|\mathcal{L}^i\right|$ for every $i \in [m]$, the last derivation holds by summing the geometric sequence, and inequalities (92) and (93) hold since $\rho \geq 1$. □

Overview of Algorithm 6 Suggested implementation for robust median; See Definition 2.5.

Algorithm 6: MEDIAN$(\mathcal{P}, k, \delta)$

Input: An (n, m)-set $\mathcal{P}$, a positive integer $k \leq 1$, and probability of failure $\delta \in (0, 1)$.
Output: A point $q \in \mathcal{X}$ that satisfies Lemma C.8
$b :=$ a universal constant that can be determined from the proof of Lemma C.8
Pick a random sample $\mathcal{S}$ of $|\mathcal{S}| = b \cdot k^2 log(\frac{1}{\delta})$ sets from $\mathcal{P}$
$q :=$ a point that minimizes $\sum_{p \in \mathrm{closest}(\mathcal{S}, \{q\}, (1-\tau)\gamma)} \tilde{D}(p, q)$ over $q \in Q \in \mathcal{S}$
Return q

Lemma C.8 (Lemma 5.1 in [26]). *Let $\mathcal{P}$ be an (n, m)-set in $\mathcal{X}$, $k \geq 1$, and $\delta \in (0, 1)$. Let $q \in \mathcal{X}$ be the output of* MEDIAN$(\mathcal{P}, k, \delta)$*; see Algorithm 6. Then, with probability at least $1 - \delta$, q is a $\left(\frac{1}{2k}, \frac{1}{6}, 2\right)$-median for $\mathcal{P}$; see Definition 2.5. Moreover , q can be computed in $O(tb^2k^4 \log(\frac{1}{\delta}))$ time, where t is the time it takes to compute $\tilde{D}(P, Q)$ for every pair $P, Q \in \mathcal{P}$.*