Coresets for Clustering in Excluded-minor Graphs and Beyond[*]

Vladimir Braverman[†] Shaofeng H.-C. Jiang[‡] Robert Krauthgamer[§] Xuan Wu[†]

Abstract

Coresets are modern data-reduction tools that are widely used in data analysis to improve efficiency in terms of running time, space and communication complexity. Our main result is a fast algorithm to construct a small coreset for k-Median in (the shortest-path metric of) an excluded-minor graph. Specifically, we give the first coreset of size that depends only on k, ϵ and the excluded-minor size, and our running time is quasi-linear (in the size of the input graph).

The main innovation in our new algorithm is that is iterative; it first reduces the n input points to roughly $O(\log n)$ reweighted points, then to $O(\log\log n)$, and so forth until the size is independent of n. Each step in this iterative size reduction is based on the importance sampling framework of Feldman and Langberg (STOC 2011), with a crucial adaptation that reduces the number of *distinct points*, by employing a terminal embedding (where low distortion is guaranteed only for the distance from every terminal to all other points). Our terminal embedding is technically involved and relies on shortest-path separators, a standard tool in planar and excluded-minor graphs.

Furthermore, our new algorithm is applicable also in Euclidean metrics, by simply using a recent terminal embedding result of Narayanan and Nelson (STOC 2019), which extends the Johnson-Lindenstrauss Lemma. We thus obtain an efficient coreset construction in high-dimensional Euclidean spaces, thereby matching and simplifying state-of-the-art results (Sohler and Woodruff, FOCS 2018; Huang and Vishnoi, STOC 2020).

In addition, we also employ terminal embedding with additive distortion to obtain small coresets in graphs with bounded highway dimension, and use applications of our coresets to obtain improved approximation schemes, e.g., an improved PTAS for planar k-Median via a new centroid set.

1 Introduction

Coresets are modern tools for efficient data analysis that have become widely used in theoretical computer science, machine learning, networking and other areas. This paper investigates coresets for the metric k-Median problem that can be defined as follows. Given an *ambient* metric space $M = (V, d)$ and a *weighted* set $X \subseteq V$ with weight function $w : X \to \mathbb{R}_+$, the goal is to find a set of k *centers* $C \subseteq V$ that minimizes the total cost of connecting every point to a center in C:

$$\mathrm{cost}(X, C) := \sum_{x \in X} w(x) \cdot d(x, C),$$

where $d(x, C) := \min_{y \in C} d(x, y)$ is the distance to the closest center. An *ϵ-coreset* for k-Median on X is a weighted subset $D \subseteq X$, such that

$$\forall C \subseteq V, |C| = k, \qquad \mathrm{cost}(D, C) \in (1 \pm \epsilon) \cdot \mathrm{cost}(X, C).$$

We note that many papers study a more general problem, (k, z)-Clustering, where inside the cost function each distance is raised to power z. We focus on k-Median for sake of exposition, but most of our results easily extend to (k, z)-Clustering.

Small coresets are attractive since one can solve the problem on D instead of X and, as a result, improve time, space or communication complexity of downstream applications [41, 42, 22]. Thus, one of the most important performance measures of a coreset D is its *size*, i.e., the number of distinct points in it, denoted $\|D\|_0$.[1] Har-Peled and Mazumdar [30] introduced the above definition and designed the first coreset for k-Median in Euclidean spaces ($V = \mathbb{R}^m$ with ℓ_2 norm),

[*] Some details are omitted due to space constraints and can be found in the full version arXiv:2004.07718.

[†] Johns Hopkins University. This research was supported in part by NSF CAREER grant 1652257, NSF grant 1934979, ONR Award N00014-18-1-2364 and the Lifelong Learning Machines program from DARPA/MTO. Email: vova@cs.jhu.edu, xwu71@jh.edu

[‡] Aalto University. Part of this work was done when the author was at Weizmann Institute of Science. Email: shaofeng.jiang@aalto.fi

[§] Weizmann Institute of Science. Work partially supported by ONR Award N00014-18-1-2364, the Israel Science Foundation grant #1086/18, and a Minerva Foundation grant. Part of this work was done while some of the authors were visiting the Simons Institute for the Theory of Computing. Email: robert.krauthgamer@weizmann.ac.il

[1] For a weighted set X, we denote by $\|X\|_0$ the number of *distinct* elements, by $\|X\|_1$ its total weight.

and since their work, designing small coresets has become a flourishing research direction, including not only k-Median and (k,z)-Clustering e.g. [29, 12, 38, 20, 51, 33, 22], but also many other important problems, such as subspace approximation/PCA [19, 21, 22], projective clustering [20, 56, 22], regression [43], density estimation [37, 47], ordered weighted clustering [10], and fair clustering [49, 32].

Many modern coreset constructions stem from a fundamental framework proposed by Feldman and Langberg [20], extending the importance sampling approach of Langberg and Schulman [38]. In this framework [20], the size of an ϵ-coreset for k-Median is bounded by $O(\text{poly}(k/\epsilon)\cdot \text{sdim})$, where sdim is the shattering (or VC) dimension of the family of distance functions. For a general metric space (V,d), a direct application of [20] results in a coreset of size $O_{k,\epsilon}(\log|V|)$, which is tight in the sense that in some instances, every coreset must have size $\Omega(\log|V|)$ [5].Therefore, to obtain coresets of size independent of the data set X, we have to restrict our attention to specific metric spaces, which raises the following fundamental question.

Question 1.1. *Identify conditions on a data set X from metric space (V,d) that guarantee the existence (and efficient construction) of an ϵ-coreset for k-*Median *of size $O_{\epsilon,k}(1)$?*

This question has seen major advances recently. Coresets of size independent of X (and V) were obtained, including efficient algorithms, for several important special cases: high-dimensional Euclidean spaces [51, 24, 33] (i.e., independently of the Euclidean dimension), metrics with bounded doubling dimension [31], and shortest-path metric of bounded-treewidth graphs [5].

1.1 Our Results

Overview We make significant progress on this front (Question 1.1) by designing new coresets for k-Median in three very different types of metric spaces. Specifically, we give (i) the first $O_{\epsilon,k}(1)$-size coreset for excluded-minor graphs; (ii) the first $O_{\epsilon,k}(1)$-size coreset for graphs with bounded highway dimension; and (iii) a simplified state-of-the-art coreset for high-dimensional Euclidean spaces (i.e., coreset-size independent of the Euclidean dimension with guarantees comparable to [33] but simpler analysis.)

Our coreset constructions are all based on the well-known importance sampling framework of [20], but with subtle deviations that introduce significant advantages. Our first technical idea is to relax the goal of computing the final coreset in one shot: we present a general reduction that turns an algorithm that computes a coreset of size $O(\text{poly}(k/\epsilon)\log\|X\|_0)$ into an algorithm that computes a coreset of size $O(\text{poly}(k/\epsilon))$. The reduction is very simple and efficient, by straightforward iterations. Thus, it suffices to construct a coreset of size $O(\text{poly}(k/\epsilon)\log\|X\|_0)$. We construct this using the importance sampling framework [20], but applied in a subtly different way, called terminal embedding, in which distances are slightly distorted, trading accuracy for (hopefully) a small shattering dimension. It still remains to bound the shattering dimension, but we are now much better equipped — we can distort the distances (design a new embedding or employ a known one), and we are content with dimension bound $O_{k,\epsilon}(\log\|X\|_0)$, instead of the usual $O_{k,\epsilon}(1)$.

We proceed to present each of our results and its context-specific background, see also Table 1 for summary, and then describe our techniques at a high-level in Section 1.2.

Coresets for Clustering in Graph Metrics k-Median clustering in graph metrics, i.e. shortest-path metric of graphs, is a central task in data mining of spatial networks (e.g., planar networks such as road networks) [50, 57], and has applications in various location optimization problems, such as placing servers on the Internet [39, 35] (see also a survey [52]), and in data analysis methods [48, 17]. We obtain new coresets for excluded-minor graphs and new coresets for graphs of bounded highway dimension. The former generalize planar graphs and the latter capture the structure of transportation networks.

Coresets for Excluded-minor Graphs A *minor* of graph G is a graph H obtained from G by a sequence of edge deletions, vertex deletions or edge contractions. We are interested in graphs G that exclude a fixed graph H as a minor, i.e., they do not contain H as a minor. Excluded-minor graphs have found numerous applications in theoretical computer science and beyond and they include, for example, planar graphs and bounded-treewidth graphs. Besides its practical importance, k-Median in planar graphs received significant attention in approximation algorithms research [54, 15, 16]. Our framework yields the first ϵ-coreset of size $O_{k,\epsilon}(1)$ for k-Median in excluded-minor graphs, see Corollary 4.1 for details. Such a bound was previously known only for the special case of bounded-treewidth graphs [5]. We stress that our technical approach is significantly different from [5]; we introduce a novel iterative construction and a relaxed terminal embedding of excluded-minor graph metrics (see Section 1.2), and overall bypass bounding the shattering dimension by $O(1)$ (which is the technical core in [5]).

Coresets for Graphs with Bounded Highway Dimension Due to the tight relation to road networks,

Table 1: our results of ϵ-coresets for k-Median in various types of metric spaces $M(V, d)$ with comparison to previous works. By graph metric, we mean the shortest-path metric of an edge-weighted graph $G = (V, E)$. Corollary 4.2 (and [33]) also work for general (k, z)-Clustering, but we list the result for k-Median ($z = 1$) only.

Metric space		Coreset size[2]	Reference
General metrics		$\tilde{O}(\epsilon^{-2}k \log \lvert V\rvert)$	[20]
Graph metrics	Bounded treewidth[3]	$\tilde{O}(\epsilon^{-2}k^2)$	[5]
	Excluding a fixed minor	$\tilde{O}(\epsilon^{-4}k^2)$	Corollary 4.1
	Bounded highway dimension	$\tilde{O}(k^{O(\log(1/\epsilon))})$	Corollary 4.3
Euclidean $\mathbb{R}^m$	Dimension-dependent	$\tilde{O}(\epsilon^{-2}km)$	[20]
	Dimension-free	$\tilde{O}(\epsilon^{-4}k)$	[33], Corollary 4.2

graphs of bounded highway dimension is another important family for the study of clustering in graph metrics. The notion of highway dimension was first proposed by [2] to measure the complexity of transportation networks such as road networks and airline networks. Intuitively, it captures the fact that going from any two far-away cities A and B, the shortest path between A and B always goes through a small number of connecting hub cities. The formal definition of highway dimension is given in Definition 4.1, and we compare related versions of definitions in Remark 4.3. The study of highway dimension was originally to understand the efficiency of heuristics for shortest path computations [2], while subsequent works also study approximation algorithms for optimization problems such as TSP, Steiner Tree [23] and k-Median [7]. We show the first coreset for graphs with bounded highway dimension, and as we will discuss later it can be applied to design new approximation algorithms. The formal statement can be found in Corollary 4.3.

Coresets for High-dimensional Euclidean Space The study of coresets for k-Median (and more generally (k, z)-Clustering) in Euclidean space $\mathbb{R}^m$ spans a rich line of research. The first coreset for k-Median in Euclidean spaces, given by [30], has size $O(k\epsilon^{-m} \log n)$ where $n = \|X\|_1$, and the $\log n$ factor was shaved by a subsequent work [29]. The exponential dependence on the Euclidean dimension m was later improved to $\text{poly}(km/\epsilon)$ [38], and to $O(km/\epsilon^2)$ [20]. Very recently, the first coreset for k-Median of size $\text{poly}(k/\epsilon)$, which is *independent* of the Euclidean dimension m,[4] was obtained by [51] (see also [24]).[5] This was recently improved in [33], which designs a (much faster) near-linear time construction for (k, z)-Clustering, with slight improvements in the coreset size and the (often useful) additional property that the coreset is a subset of X. Our result extends this line of research; an easy application of our new framework yields a near-linear time construction of coreset of size $\text{poly}(k/\epsilon)$, which too is independent of the dimension m. Compared to the state of the art [33], our result achieves essentially the same size bound, while greatly simplifying the analysis. A formal statement and detailed comparison with [33] can be found in Corollary 4.2 and Remark 4.2.

Applications: Improved Approximation Schemes We apply our coresets to design approximation schemes for k-Median in shortest-path metrics of planar graphs and graphs with bounded highway dimension. In particular, we give an FPT-PTAS, parameterized by k and ϵ, in graphs with bounded highway dimension, and a PTAS in planar graphs. Both algorithms run in time near-linear in $|V|$, and improve previous results in the corresponding settings.

The PTAS for k-Median in planar graphs is obtained using a new centroid-set result. A *centroid set* is a subset of V that contains centers giving a $(1 + \epsilon)$-approximate solution. We obtain centroid sets of size *independent* of the input X in planar graphs, which improves a recent size bound $(\log |V|)^{O(1/\epsilon)}$ [16], and moreover runs in time near-linear in $|V|$.

Due to the space limit, details of these applications are omitted and they can be found in the full version.

[2] Throughout, the notation $\tilde{O}(f)$ hides $\text{poly} \log f$ factors, and $O_m(f)$ hides factors that depend on m.

[3] In fact, the main claim in [5] was a weaker bound of $\tilde{O}(\epsilon^{-2}k^3)$, but it was noted that the dependence in k may be reduced to k^2 by using an improved framework in a recent work [22].

[4] Dimension-independent coresets were obtained earlier for Euclidean k-Means [9, 22], however these do not apply to k-Median.

[5] The focus of [51] is on k-Median, but the results extend to (k, z)-Clustering.

1.2 Technical Contributions

Iterative Size Reduction This technique is based on an idea so simple that it may seem too naive: Basic coreset constructions have size $O_{k,\epsilon}(\log n)$, so why not apply it repeatedly, to obtain a coreset of size $O_{k,\epsilon}(\log\log n)$, then $O_{k,\epsilon}(\log\log\log n)$ and so on? One specific example is the size bound $O(\epsilon^{-2}k\log n)$ for a general n-point metric space [20], where this does not work because $n = |V|$ is actually the size of the *ambient* space, irrespective of the *data* set X. Another example is the size bound $O(\epsilon^{-m}k\log n)$ for Euclidean space $\mathbb{R}^m$ [30], where this does not work because $n = \|X\|_1$ is the total weight of the data points X, which coresets do not reduce (to the contrast, they maintain it). These examples suggest that one should avoid two pitfalls: dependence on V and dependence on the total weight.

We indeed make this approach work by requiring an algorithm $\mathcal{A}$ that constructs a coreset of size $O(\log\|X\|_0)$, which is *data-dependent* (recall that $\|X\|_0$ is the number of *distinct* elements in a weighted set X). Specifically, we show in Theorem 3.1 that, given an algorithm $\mathcal{A}$ that constructs an ϵ'-coreset of size $O(\text{poly}(k/\epsilon')\log\|X\|_0)$ for every ϵ' and $X \subseteq V$, one can obtain an ϵ-coreset of size poly(k/ϵ) by simply applying $\mathcal{A}$ iteratively. It follows by setting ϵ' carefully, so that it increases quickly and eventually $\epsilon' = O(\epsilon)$. See Section 3.1 for details.

Not surprisingly, the general idea of applying the sketching/coreset algorithm iteratively was also used in other related contexts (e.g. [40, 13, 45]). Moreover, a related two-step iterative construction was applied in a recent coreset result [33]. Nevertheless, the exact implementation of iterative size reduction in coresets is unique in the literature. As can be seen from our results, this reduction fundamentally helps to achieve new or simplified coresets of size *independent* of data set. We expect the iterative size reduction to be of independent interest to future research.

Terminal Embeddings To employ the iterative size reduction, we need to construct coresets of size poly$(k/\epsilon)\cdot\log\|X\|_0$. Unfortunately, a direct application of [20] yields a bound that depends on the number of vertices $|V|$, irrespective of X. To bypass this limitation, the framework of [20] is augmented (in fact, we use a refined framework proposed in [22]), to support controlled modifications to the distances $d(\cdot,\cdot)$. As explained more formally in Section 3.2, one represents these modifications using a set of functions $\mathcal{F} = \{f_x : V \to \mathbb{R}_+ \mid x \in X\}$, that corresponds to the modified distances from each x, i.e., $f_x(\cdot) \leftrightarrow d(x,\cdot)$. Many previous papers [38, 20, 9, 22] work directly with the distances and use the function set $\mathcal{F} = \{f_x(\cdot) = d(x,\cdot) \mid x \in X\}$, or a more sophisticated but still direct variant of hyperbolic balls (where each f_x is an affine transformation of $d(x,\cdot)$). A key difference is that we use a "proxy" function set $\mathcal{F}$, where each $f_x(\cdot) \approx d(x,\cdot)$. This introduces a tradeoff between the approximation error (called distortion) and the shattering dimension of $\mathcal{F}$ (which controls the number of samples), and overall results in a smaller coreset. Such tradeoff was first used in [31] to obtain small coresets for doubling spaces, and was recently used in [33] to reduce the coreset size for Euclidean spaces. This proxy function set may be alternatively viewed as a *terminal embedding* on X, in which both the distortion of distances (between X and all of V) and the shattering dimension are controlled.

We then consider two types of terminal embeddings $\mathcal{F}$. The first type (Section 3.3) maintains $(1+\epsilon)$-multiplicative distortion of the distances. When this embedding achieves dimension bound $O(\text{poly}(k/\epsilon)\log\|X\|_0)$, we combine it with the aforementioned iterative size reduction, to further reduce the size to be independent of X. It remains to actually design embeddings of this type, which we achieve (as explained further below), for excluded-minor graphs and for Euclidean spaces, and thus we overall obtain $O_{\epsilon,k}(1)$-size coresets in both settings. Our second type of terminal embeddings $\mathcal{F}$ (Section 3.4) maintains additive distortion on top of the multiplicative one. We design embeddings of this type (as explained further below) for graphs with bounded highway dimension; these embeddings have shattering dimension poly(k/ϵ), and thus we overall obtain $O_{\epsilon,k}(1)$-size coresets even without the iterative size reduction. We report our new terminal embeddings in Table 2.

Terminal Embedding for Euclidean Spaces Our terminal embedding for Euclidean spaces is surprisingly simple, and is a great showcase for our new framework. In a classical result [20], it has been shown that $\text{sdim}_{\max}(\mathcal{F}) = O(m)$ for Euclidean distance in $\mathbb{R}^m$ without distortion. On the other hand, we notice a terminal embedding version of Johnson-Lindenstrauss Lemma was discovered recently [46]. Our terminal embedding bound (Lemma 4.8) follows by directly combining these two results, see Section 4.3 for details.

We note that without our iterative size reduction technique, plugging in the recent terminal Johnson-Lindenstrauss Lemma [46] into classical importance sampling frameworks, such as [20, 22] does not yield any interesting coreset. Furthermore, the new terminal Johnson-Lindenstrauss Lemma was recently used in [33] to design coresets for high-dimensional Euclidean spaces. Their size bounds are essentially the same as ours, however they go through a complicated analysis to directly show a shattering dimension bound poly(k/ϵ). This complication is not necessary in our method, be-

Table 2: New terminal embeddings $\mathcal{F}$ for different metrics spaces. The reported distortion bound is the upper bound on $f_x(c)$, in addition to the lower bound $f_x(c) \geq d(x, c)$. The embeddings of graphs with bounded highway dimension, called here "highway graphs" for short, are defined with respect to a given $S \subseteq V$ (see Lemma 4.9).

Metric space	Dimension $\mathrm{sdim}_{\max}(\mathcal{F})$	Distortion	Result
Euclidean	$O(\epsilon^{-2} \log \|X\|_0)$	$(1+\epsilon) \cdot d(x, c)$	Lemma 4.8
Excluded-minor graphs	$\tilde{O}(\epsilon^{-2} \log \|X\|_0)$	$(1+\epsilon) \cdot d(x, c)$	Lemma 4.1
Highway graphs	$O(\|S\|^{O(\log(1/\epsilon))})$	$(1+\epsilon) \cdot d(x, c) + \epsilon \cdot d(x, S)$	Lemma 4.9

cause by our iterative size reduction it suffices to show a very loose $O_{k,\epsilon}(\log \|X\|_0)$ dimension bound, and this follows immediately from the Johnson-Lindenstrauss result.

Terminal Embedding for Excluded-minor Graphs The technical core of the terminal embedding for excluded-minor graphs is how to bound the shattering dimension. In our proof, we reduce the problem of bounding the shattering dimension into finding a representation of the distance functions on $X \times V$ as a set of *min-linear* functions. Specifically, we need to find for each x a min-linear function $g_x : \mathbb{R}^s \to \mathbb{R}$ of the form $g_x(t) = \min_{1 \leq i \leq s}\{a_i t_i + b_i\}$, where $s = O(\log \|X\|_0)$, such that $\forall c \in V$, there is $t \in \mathbb{R}^s$ with $d(x, c) = g_x(t)$.

The central challenge is how to relate the graph structure to the structure of shortest paths $d(x, c)$. To demonstrate how we relate them, we start with discussing the simple special case of bounded treewidth graphs. For bounded treewidth graphs, the vertex separator theorem is applied to find a subset $P \subseteq V$, through which the shortest path $x \rightsquigarrow y$ has to pass. This translates into the following

$$d(x, c) = \min_{p \in P}\{d(x, p) + d(p, c)\},$$

and for each $x \in X$, we can use this to define the desired min-linear function $g_x(d(p_1, c), \ldots, d(p_m, c)) = d(x, c)$, where we write $P = \{p_1, \ldots, p_m\}$.

However, excluded-minor graphs do not have small vertex separator, and we use the shortest-path separator [53, 3] instead. Now assume for simplicity that the shortest paths $x \rightsquigarrow c$ all pass through a fixed shortest path l. Because l itself is a shortest path, we know $\forall x \in X, c \in V$,

$$d(x, c) = \min_{u_1, u_2 \in l}\{d(x, u_1) + d(u_1, u_2) + d(u_2, c)\}.$$

Since l can have many (i.e. $\omega(\log \|X\|_0)$) points, we need to discretize l by designating $\mathrm{poly}(\epsilon^{-1})$ *portals* P_x^l on l for each $x \in X$ (and similarly P_c^l for $c \in V$). This only introduces $(1+\epsilon)$ distortion to the distance, which we can afford.

Then we create $d'_x : l \to \mathbb{R}_+$ to approximate $d(x, u)$'s, using distances from x to the portals P_x^l (and similarly for $d(c, u)$). Specifically, for the sake of presentation, assume $P_x^l = \{p_1, p_2, p_3\}$ ($p_1 \leq p_2 \leq p_3$), interpret l as interval $[0, 1)$, then for $u \in [0, p_1)$, define $d'_x(u) = d(x, 0)$, for $u \in [p_1, p_2)$, define $d'_x(u) = d(x, p_1)$, and so forth. Hence, each $d'_x(\cdot)$ is a piece-wise linear function of $O(|P_x^l|)$ pieces (again, similarly for $d'_c(\cdot)$), and this enables us to write $d(x, c) \approx d'(x, c)$, where

$$d'(x, c) := \min_{u_1, u_2 \in P_x^l \cup P_c^l}\{d'_x(u_1) + d(u_1, u_2) + d'_c(u_2)\}.$$

Therefore, it suffices to find a min-linear representation for $d'(x, \cdot)$ for $x \in X$. However, the piece-wise linear structure of d'_x creates extra difficulty to define min-linear representations. To see this, still assume $P_x^l = \{p_1, p_2, p_3\}$. Then to determine $d'_x(u)$ for $u \in P_x^l \cup P_c^l$, we not only need to know $d(x, p_i)$ for $p_i \in P_x^l$, but also need to know which sub-interval $[p_i, p_{i+1})$ that u belongs to. (That is, if $u \in [p_1, p_2)$, then $d'_x(u) = d(x, p_1)$.) Hence, in addition to using distances $\{c\} \times P_c^l$ as variables of g_x, the relative ordering between points in $P_x^l \cup P_c^l$ is also necessary to evaluate $d'(x, c)$.

Because $c \in V$ can be arbitrary, we cannot simply "remember" the ordering in g_x. Hence, we "guess" this ordering, and for each fixed ordering we can write g_x as a min-linear function of few variables. Luckily, we can afford the "guess" since $|P_x^l \cup P_c^l| = \mathrm{poly}(\epsilon^{-1})$ which is independent of X. A more detailed overview can be found in Section 4.1.

Terminal Embedding for Graphs with Bounded Highway Dimension In addition to a $(1+\epsilon)$ multiplicative error, the embedding for graphs with bounded highway dimension also introduces an additive error. In particular, for a given $S \subseteq V$, it guarantees that $\forall x \in X, c \in V$

$$d(x, c) \leq f_x(c) \leq (1+\epsilon) \cdot d(x, c) + \epsilon \cdot d(x, S).$$

This terminal embedding is a direct consequence of a similar embedding from graphs with bounded highway dimension to graphs with bounded treewidth [7], and a previous result about the shattering dimension for graphs with bounded treewidth [5]. In our applications, we will choose S to be a constant approximate solution[6] $C^\star$ to k-Median. So the additive error becomes $\epsilon \cdot d(x, C^\star)$. In general, this term can still be much larger than $d(x,c)$, but the *collectively* error in the clustering objective is bounded. This observation helps us to obtain a coreset, and due to the additional additive error, the shattering dimension is already independent of X and hence no iterative size reduction is necessary.

1.3 Related Work

Approximation algorithms for metric k-Median have been extensively studied. In general metric spaces, it is NP-hard to approximate k-Median within a $1 + \frac{2}{e}$ factor [34], and the state of the art is a $(2.675 + \epsilon)$-approximation [11]. In Euclidean space $\mathbb{R}^m$, k-Median is APX-hard if both k and the dimension m are part of the input [27]. However, PTAS's do exist if either k or dimension m is fixed [30, 4, 15, 25].

Tightly related to coresets, dimensionality reduction has also been studied for clustering in Euclidean spaces. Compared with coresets which reduce the data set size while keeping the dimension, dimensionality reduction aims to find a low-dimensional representation of data points (but not necessarily reduce the number of data points). As a staring point, a trivial application of Johnson-Lindenstrauss Lemma [36] yields a dimension bound $O(\epsilon^{-2} \log n)$ for (k,z)-Clustering. For k-Means with $1 + \epsilon$ approximation ratio, [14] showed an $O(k/\epsilon^2)$ dimension bound for data-oblivious dimension reduction and an $O(k/\epsilon)$ bound for the data-dependent setting. Moreover, the same work [14] also obtained a data-oblivious $O(\epsilon^{-2} \log k)$ dimension bound for k-Means with approximation ratio $9 + \epsilon$. Very recently, [6] obtained an $\tilde{O}(\epsilon^{-6}(\log k + \log\log n))$ dimension bound for k-Means and [44] obtained an $O(\epsilon^{-2} \log \frac{k}{\epsilon})$ bound for (k,z)-Clustering. Both of them used data-oblivious methods and have approximation ratio $1+\epsilon$. Dimensionality reduction techniques are also used for constructing dimension-free coresets in Euclidean spaces [51, 6, 33, 22].

2 Preliminaries

Notations Let $V^k := \{C \subseteq V : |C| \leq k\}$ denote the collection of all subsets of V of size at most k. [7] For integer $n, i > 0$, let $\log^{(i)} n$ denote the i-th iterated logarithm of n, i.e. $\log^{(1)} n := \log n$ and $\log^{(i)} n := \log(\log^{(i-1)} n)$ $(i \geq 2)$. Define $\log^\star n$ as the number of times the logarithm is iteratively applied before the result is at most 1, i.e. $\log^\star n := 0$ if $n \leq 1$ and $\log^\star n = 1 + \log^\star(\log n)$ if $n > 1$. For a weighted set S, denote the weight function as $w_S : S \to \mathbb{R}_+$. Let $\mathrm{OPT}_z(X)$ be the optimal objective value for (k,z)-Clustering on X, and we call a subset $C \subseteq V$ an (α, β)-approximate solution for (k,z)-Clustering on X if $|C| = \alpha k$ and $\mathrm{cost}_z(X, C) := \sum_{x \in X} w_X(x) \cdot (d(x,C))^z \leq \beta \cdot \mathrm{OPT}_z(X)$.

Functional Representation of Distances We consider sets of functions $\mathcal{F}$ from V to $\mathbb{R}_+$. Specifically, we consider function sets $\mathcal{F} = \{f_x : V \to \mathbb{R}_+ \mid x \in X\}$ that is indexed by the weighted data set $X \subseteq V$, and intuitively $f_x(\cdot)$ is used to measure the distance from $x \in X$ to a point in V. Because we interpret f_x's as distances, for a subset $C \subseteq V$, we define $f_x(C) := \min_{c \in C} f_x(C)$, and define the clustering objective accordingly as

$$\mathrm{cost}_z(\mathcal{F}, C) := \sum_{f_x \in \mathcal{F}} w_{\mathcal{F}}(f_x) \cdot (f_x(C))^z.$$

In fact, in our applications, we will use $f_x(y)$ as a "close" approximation to d. We note that this functional representation is natural for k-Clustering, since the objective function only uses distances from X to every k-subset of V only. Furthermore, we do not require the triangle inequality to hold for such functional representations.

Shattering Dimension For $c \in V, r \geq 0$, define $B_{\mathcal{F}}(c,r) := \{f \in \mathcal{F} : f(c) \leq r\}$. We emphasize that c is from the ambient space V in addition to the data set X. Intuitively, $B_{\mathcal{F}}(c,r)$ is the ball centered at c with radius r when the f functions are used to measure distances. For example, consider $X = V$ and let $f_x(\cdot) := d(x, \cdot)$ for $x \in V$. Then $B_{\mathcal{F}}(c,r) = \{f_x \in \mathcal{F} : d(c,x) \leq r\}$, which corresponds to the metric ball centered at c with radius r.

We introduce the notion of shattering dimension in Definition 2.1. In fact, the shattering dimension may be defined with respect to any set system [28], but we do not need this generality here and thus we consider only the shattering dimension of the "metric balls" system.

Definition 2.1. (Shattering Dimension [28]) *Suppose $\mathcal{F}$ is a set of functions from V to $\mathbb{R}_+$. The shattering dimension of $\mathcal{F}$, denoted as* $\mathrm{sdim}(\mathcal{F})$*, is the smallest integer t, such that for every $\mathcal{H} \subseteq \mathcal{F}$ with $|\mathcal{H}| \geq 2$,*

(2.1)
$$\forall \mathcal{H} \subseteq \mathcal{F}, |\mathcal{H}| \geq 2, \quad |\{B_{\mathcal{H}}(c,r) : c \in V, r \geq 0\}| \leq |\mathcal{H}|^t.$$

The shattering dimension is tightly related to the well-known VC-dimension [55], and they are equal to

[6] in fact, a bi-criteria approximation suffices.

[7] Strictly speaking, V^k is the collection of all ordered k-tuples of V, but here we use it to denote the subsets. Note that tuples may contain repeated elements so the subsets in V^k are of size at most k.

each other up to a logarithmic factor [28, Corollary 5.12, Lemma 5.14]. In our application, we usually do not use $\mathrm{sdim}(\mathcal{F})$ directly. Instead, given a point weight $v : X \to \mathbb{R}_+$, we define $\mathcal{F}_v := \{f_x \cdot v(x) \mid x \in X\}$, and then consider the maximum of $\mathrm{sdim}(\mathcal{F}_v)$ over all possible v, defined as $\mathrm{sdim}_{\max}(\mathcal{F}) := \max_{v:X\to\mathbb{R}_+} \mathrm{sdim}(\mathcal{F}_v)$.

3 Framework

We present our general framework for constructing coresets. Our first new idea is a generic reduction, called iterative size reduction, through which it suffices to find a coreset of size $O(\log \|X\|_0)$ only in order to get a coreset of size independent of X. This general reduction greatly simplifies the coreset construction, and in particular, as we will see, "old" techniques such as importance sampling gains new power and becomes useful for new settings such as excluded-minor graphs.

Roughly speaking, the iterative size reduction turns a coreset construction algorithm $\mathcal{A}(X, \epsilon)$ with size $O(\mathrm{poly}(\epsilon^{-1}k) \cdot \log \|X\|_0)$ into a construction $\mathcal{A}'(X, \epsilon)$ with size $\mathrm{poly}(\epsilon^{-1}k)$. To define $\mathcal{A}'$, we simply iteratively apply $\mathcal{A}$, i.e. $X_i := \mathcal{A}(X_{i-1}, \epsilon_i)$, and terminate when $\|X_i\|_0$ does not decrease. However, if $\mathcal{A}$ is applied for t times in total, the error of the resulted coreset is accumulated as $\sum_{i=1}^{t} \epsilon_t$. Hence, to make the error bounded, we make sure $\epsilon_i \geq 2\epsilon_{i-1}$ and $\epsilon_t = O(\epsilon)$, so $\sum_{i=1}^{t} \epsilon_i = O(\epsilon)$. Moreover, our choice of ϵ_i also guarantees that $\|X_i\|_0$ is roughly $\mathrm{poly}(\epsilon^{-1}k \cdot \log^{(i)} \|X\|_0)$. Since $\log^{(i)} \|X\|_0$ decreases very fast with respect to i, $\|X_i\|_0$ becomes $\mathrm{poly}(\epsilon^{-1}k)$ in about $t = \log^\star \|X\|_0$ iterations. The detailed algorithm $\mathcal{A}'$ can be found in Algorithm 1, and we present the formal analysis in Theorem 3.1.

To construct the actual coresets which is to be used with the reduction, we adapt the importance sampling method that was proposed by Feldman and Langberg [20]. In previous works, the size of the coresets from importance sampling is related to the shattering dimension of metric balls system (i.e. in our language, it is the shattering dimension of $\mathcal{F} = \{d(x, \cdot) \mid x \in X\}$.) Instead of considering the metric balls only, we give a generalized analysis where we consider a general set of "distance functions" $\mathcal{F}$ that has some error but is still "close" to d. The advantage of doing so is that we could trade the accuracy with the shattering dimension, which in turn reduces the size of the coreset.

We particularly examine two types of such functions $\mathcal{F} = \{f_x : V \to \mathbb{R}_+ \mid x \in X\}$. The first type $\mathcal{F}$ introduces a multiplicative $(1 + \epsilon)$ error to d, i.e. $\forall x \in X, c \in V$, $d(x, c) \leq f_x(c) \leq (1 + \epsilon) \cdot d(x, c)$. Such a small distortion is already very helpful to obtain an $O(\log \|X\|_0)$ shattering dimension for minor-free graphs and Euclidean spaces. In addition to the multiplicative error, the other type of $\mathcal{F}$ introduces a certain additive error, and we make use of this to show $O(k)$ shattering dimension bound for bounded highway dimension graphs and doubling spaces. In this section, we will discuss how the two types of function sets imply efficient coresets, and the dimension bounds for various metric families will be analyzed in Section 4 where we also present the coreset results.

3.1 Iterative Size Reduction

Theorem 3.1. (Iterative Size Reduction) *Let $\rho \geq 1$ be a constant and let $\mathcal{M}$ be a family of metric spaces. Assume $\mathcal{A}(X, k, z, \epsilon, \delta, M)$ is a randomized algorithm that constructs an ϵ-coreset of size $\epsilon^{-\rho} s(k) \log \delta^{-1} \log \|X\|_0$ for (k, z)-*Clustering *on every weighted set $X \subseteq V$ and metric space $M(V, d) \in \mathcal{M}$, for every $z \geq 1, 0 < \epsilon, \delta < \frac{1}{4}$, running in time $T(\|X\|_0, k, z, \epsilon, \delta, M)$ with success probability $1 - \delta$. Then algorithm $\mathcal{A}'(X, k, z, \epsilon, \delta, M)$, stated in Algorithm 1, computes an ϵ-coreset of size $\tilde{O}(\epsilon^{-\rho} s(k) \log \delta^{-1})$ for (k, z)-*Clustering *on every weighted set $X \subseteq V$ and metric space $M(V, d) \in \mathcal{M}$, for every $z \geq 1$, $0 < \epsilon, \delta < \frac{1}{4}$, in time*

$$O\left(T\big(\|X\|_0, k, z, \frac{\epsilon}{(\log \|X\|_0)^{\frac{1}{\rho}}}, \frac{\delta}{\|X\|_0}, M\big) \cdot \log^\star \|X\|_0\right),$$

and with success probability $1 - \delta$.

Algorithm 1 Iterative size reduction $\mathcal{A}'(X, k, z, \epsilon, \delta, M)$

```
Require: algorithm $\mathcal{A}(X, k, z, \epsilon, \delta, M)$ that computes an $\epsilon$-coreset for $(k, z)$-Clustering on $X$ with size $\epsilon^{-\rho} s(k) \log \delta^{-1} \log \|X\|_0$ and success probability $1 - \delta$.
let $X_0 := X$, and let $t$ be the largest integer such that $\log^{(t-1)} \|X\|_0 \geq \max\{20\epsilon^{-\rho} s(k) \log \delta^{-1}, \rho 2^{\rho+1}\}$
for $i = 1, \cdots, t$ do
    let $\epsilon_i := \epsilon / (\log^{(i)} \|X\|_0)^{\frac{1}{\rho}}$, $\delta_i := \delta / \|X_{i-1}\|_0$
    let $X_i := \mathcal{A}(X_{i-1}, k, z, \epsilon_i, \delta_i, M)$
end for
$X_{t+1} := \mathcal{A}(X_t, k, z, \epsilon, \delta, M)$
return $X_{t+1}$
```

3.2 Importance Sampling

We proceed to design the algorithm $\mathcal{A}$ required by Theorem 3.1. It is based on the importance sampling algorithm introduced by [38, 20], and at a high level consists of two steps:

1. Computing probabilities: for each $x \in X$, compute $p_x \geq 0$ such that $\sum_{x \in X} p_x = 1$.

2. Sampling: draw N (to be determined later) independent samples from X, each drawn from the distribution $(p_x : x \in X)$, and assign each sample x a weight $\frac{w_X(x)}{p_x \cdot N}$ to form a coreset D.

The key observation in the analysis of this algorithm is that the sample size N, which is also the coreset size $\|D\|_0$, is related to the shattering dimension (see Definition 2.1) of a suitably defined set of functions [20, Theorem 4.1]. The analysis in [20] has been subsequently improved [9, 22], and we make use of [22, Theorem 31], restated as follows.

Lemma 3.1. (Importance Sampling [22]) *Fix $z \geq 1$, $0 < \epsilon < \frac{1}{2}$, an integer $k \geq 1$ and a metric space (V, d). Let $X \subseteq V$ have weights $w_X : V \to \mathbb{R}_+$ and let $\mathcal{F} := \{f_x : V \to \mathbb{R}_+ \mid x \in X\}$ be a corresponding set of functions with weights $w_{\mathcal{F}}(f_x) = w_X(x)$. Suppose $\{\sigma_x\}_{x \in X}$ satisfies*

$$\forall x \in X, \quad \sigma_x \geq \sigma_x^{\mathcal{F}} := \max_{C \in V^k} \frac{w_X(x) \cdot (f_x(C))^z}{\mathrm{cost}_z(\mathcal{F}, C)},$$

and set a suitable

$$N = O(\epsilon^{-2} \sigma_X (k \cdot \mathrm{Dim} \cdot \log(\mathrm{Dim}) \cdot \log \sigma_X + \log \tfrac{1}{\delta})),$$

where $\sigma_X := \sum_{x \in X} \sigma_x$ and

$$\mathrm{Dim} = \mathrm{sdim}_{\max}(\mathcal{F}) := \max_{v : X \to \mathbb{R}_+} \mathrm{sdim}\,(\mathcal{F}_v)$$
$$\mathcal{F}_v := \{f_x \cdot v(x) \mid x \in X\}.$$

Then the weighted set D of size $\|D\|_0 = N$ returned by the above importance sampling algorithm satisfies, with high probability $1 - \delta$,

$$\forall C \in V^k, \quad \sum_{x \in D} w_D(x) \cdot (f_x(C))^z \in (1 \pm \epsilon) \cdot \mathrm{cost}_z(\mathcal{F}, C).$$

Remark 3.1. *We should explain how [22, Theorem 31] implies Lemma 3.1. First of all, the bound in [22] is with respect to VC-dimension, and we transfer to shattering dimension by losing a logarithmic factor (see Section 2 for the relation between VC-dimension and shattering dimension). Another main difference is that the functions therein are actually not from V to $\mathbb{R}_+$. For $\mathcal{F} = \{f_x : V \to \mathbb{R}_+ \mid x \in X\}$, they consider $\mathcal{F}^k := \{f_x(C) = \min_{c \in C}\{f_x(c)\} \mid x \in X\}$, and their bound on the sample size is*

$$N = \tilde{O}(\epsilon^{-2} \sigma_X (\mathrm{sdim}_{\max}(\mathcal{F}^k) \cdot \log \sigma_X + \log \tfrac{1}{\delta})).$$

The notion of balls and shattering dimension they use (for $\mathcal{F}^k$) is the natural extension of our Definition 2.1 (from functions on V to functions on V^k), where a ball around $C \in V^k$ is $B_{\mathcal{F}}(C, r) = \{f_x \in \mathcal{F} : f_x(C) \leq r\}$, and (2.1) is replaced by

$$\left|\{B_{\mathcal{H}}(C, r) : C \in V^k, r \geq 0\}\right| \leq |\mathcal{H}|^t.$$

Our Lemma 3.1 follows from [22, Theorem 31] by using the fact $\mathrm{sdim}(\mathcal{F}^k) \leq k \cdot \mathrm{sdim}(\mathcal{F})$ *from [20, Lemma 6.5].*

Terminal Embeddings. As mentioned in Section 1, $\mathcal{F}$ in Lemma 3.1 corresponds to the distance function d, i.e., $f_x(\cdot) = d(x, \cdot)$, and Lemma 3.1 is usually applied directly to the distances, i.e., on a function set $\mathcal{F} = \{f_x(\cdot) = d(x, \cdot) \mid x \in X\}$. In our applications, we instead use Lemma 3.1 with a "proxy" function set $\mathcal{F}$ that is viewed as a *terminal embedding* on X, in which both the distortion of distances (between X and all of V) and the shattering dimension are controlled.

We consider two types of terminal embeddings $\mathcal{F}$. The first type (Section 3.3) maintains $(1+\epsilon)$-multiplicative distortion of the distances, and achieves dimension bound $O(\mathrm{poly}(k/\epsilon) \log \|X\|_0)$, and the other type of $\mathcal{F}$ (Section 3.4) maintains additive distortion on top of the multiplicative one, but then the dimension is reduced to $\mathrm{poly}(k/\epsilon)$. In what follows, we discuss how each type of terminal embedding is used to construct coresets.

3.3 Coresets via Terminal Embedding with Multiplicative Distortion

The first type of terminal embedding distorts distances between V and X multiplicatively, i.e.,

$$\forall x \in X, c \in V, \qquad d(x, c) \leq f_x(c) \leq (1 + \epsilon)\ d(x, c). \tag{3.2}$$

This natural guarantee works very well for (k, z)-Clustering in general. In particular, using such $\mathcal{F}$ in Lemma 3.1, our importance sampling algorithm will produce (with high probability) an $O(z\epsilon)$-coreset for (k, z)-Clustering.

Sensitivity Estimation. To compute a coreset using Lemma 3.1 we need to define, for every $x \in X$,

$$\sigma_x \geq \sigma_x^{\mathcal{F}} = \max_{C \in V^k} \frac{w_X(x) \cdot (f_x(C))^z}{\mathrm{cost}_z(\mathcal{F}, C)}.$$

The quantity $\sigma_x^{\mathcal{F}}$, usually called the *sensitivity* of point $x \in X$ with respect to $\mathcal{F}$ [38, 20]; essentially measures the maximal contribution of x to the clustering objective over all possible centers $C \subseteq V$. Since $f_x(y)$ approximates $d(x, y)$ by (3.2), it actually suffices to estimate the sensitivity with respect to d instead of $\mathcal{F}$, given by

$$\sigma_x^{\star} := \max_{C \in V^k} \frac{w_X(x) \cdot (d(x, C))^z}{\mathrm{cost}_z(X, C)}. \tag{3.3}$$

Even though computing $\sigma_x^\star$ exactly seems computationally difficult, we show next (in Lemma 3.2) that a good estimate can be efficiently computed given an $(O(1), O(1))$-approximate clustering. A weaker version of this lemma was presented in [56] for the case where X has unit weights, and we extend it to X with general weights. We will need the following notation. Given a subset $C \subseteq V$, denote the nearest neighbor of $x \in X$, i.e., the point in C closest to x with ties broken arbitrarily, by $\mathrm{NN}_C(x) := \arg\min\{d(x,y) : y \in C\}$. The tie-breaking guarantees that every x has a unique nearest neighbor, and thus $\mathrm{NN}_C(.)$ partitions X into $|C|$ subsets. The *cluster of x under C* is then defined as $C(x) := \{x' \in X : \mathrm{NN}_C(x') = \mathrm{NN}_C(x)\}$.

Lemma 3.2. *Fix $z \geq 1$, an integer $k \geq 1$, and a weighted set X. Given $C^{\mathrm{apx}} \in V^k$ that is an (α, β)-approximate solution for (k,z)-*Clustering *on X, define for every $x \in X$,*

$$\sigma_x^{\mathrm{apx}} := w_X(x) \cdot \left(\frac{(d(x, C^{\mathrm{apx}}))^z}{\mathrm{cost}_z(X, C^{\mathrm{apx}})} + \frac{1}{w_X(C^{\mathrm{apx}}(x))} \right).$$

Then $\sigma_x^{\mathrm{apx}} \geq \Omega(\sigma_x^\star/(\beta 2^{2z}))$ for all $x \in X$, and $\sigma_X^{\mathrm{apx}} := \sum_{x \in X} \sigma_x^{\mathrm{apx}} \leq 1 + \alpha k$.

Conclusion. Our importance sampling algorithm for this type of terminal embedding is listed in Algorithm 2. By a direct combination of Lemma 3.1 and Lemma 3.2, we conclude that the algorithm yields a coreset, which is stated formally in Lemma 3.3.

Algorithm 2 Coresets for (k,z)-Clustering for $\mathcal{F}$ with multiplicative distortion

```
1: compute an $(O(1), O(1))$-approximate solution $C^{\mathrm{apx}}$ for $(k,z)$-Clustering on $X$
2: for each $x \in X$, let $\sigma_x := w_X(x) \cdot \left( \frac{(d(x,C^{\mathrm{apx}}))^z}{\mathrm{cost}_z(X,C^{\mathrm{apx}})} + \frac{1}{w_X(C^{\mathrm{apx}}(x))} \right)$    ▷ as in Lemma 3.2
3: for each $x \in X$, let $p_x := \frac{\sigma_x}{\sum_{y \in X} \sigma_y}$
4: $N := O\left(\epsilon^{-2} 2^{2z} k \cdot \left(zk \log k \cdot \mathrm{sdim}_{\max}(\mathcal{F}) + \log \frac{1}{\delta}\right)\right)$
5: draw $N$ independent samples from $X$, each from the distribution $(p_x : x \in X)$    ▷ $\mathrm{sdim}_{\max}$ as in Lemma 3.1
6: let $D$ be the set of samples, and assign each $x \in D$ a weight $w_D(x) := \frac{w_X(x)}{p_x N}$
7: return the weighted set $D$
```

Lemma 3.3. *Fix $0 < \epsilon, \delta < \frac{1}{2}$, $z \geq 1$, an integer $k \geq 1$, and a metric space $M(V,d)$. Given a weighted set $X \subseteq V$ and respective $\mathcal{F} = \{f_x : V \to \mathbb{R}_+ \mid x \in X\}$ such that*

$$\forall x \in X, c \in V, \qquad d(x,c) \leq f_x(c) \leq (1+\epsilon) \cdot d(x,c),$$

Algorithm 2 computes a weighted set $D \subseteq X$ of size

$$\|D\|_0 = O\left(\epsilon^{-2} 2^{2z} k \left(zk \log k \cdot \mathrm{sdim}_{\max}(\mathcal{F}) + \log \tfrac{1}{\delta}\right)\right),$$

*that with high probability $1-\delta$ is an ϵ-coreset for (k,z)-*Clustering *on X.*

The running time of Algorithm 2 is dominated by the sensitivity estimation, especially line 1 which computes an $(O(1), O(1))$-approximate solution. In Lemma 3.4 we present efficient implementations of the algorithm, both in metric settings and in graph settings.

Lemma 3.4. *Algorithm 2 can be implemented in time $\tilde{O}(k\|X\|_0)$ if it is given oracle access to the distance d, and it can be implemented in time $\tilde{O}(|E|)$ if the input is an edge-weighted graph $G = (V,E)$ and M is its shortest-path metric.*

3.4 Coresets via Terminal Embedding with Additive Distortion

The second type of embedding has, in addition to the above $(1+\epsilon)$-multiplicative distortion, also an additive distortion. Specifically, we assume the function set $\mathcal{F} = \mathcal{F}_S$ is defined with respect to some subset $S \subseteq V$ and satisfies $\forall x \in X, c \in V$,

$$d(x,c) \leq f_x(c) \leq (1+\epsilon) \cdot d(x,c) + \epsilon \cdot d(x,S).$$

The important sampling algorithm for this case is largely similar to Algorithm 2, except for a slightly larger number of samples N and some hidden constants. Here, we use the embedding with S being an $(O(1), O(1))$-approximate solution, and we choose $N := O\left(\epsilon^{-2} k \left(k \log k \cdot \mathrm{sdim}_{\max}(\mathcal{F}_{C^{\mathrm{apx}}}) + \log \frac{1}{\delta}\right) + k^2 \log \frac{1}{\delta}\right)$, where $\mathcal{F}_{C^{\mathrm{apx}}}$ is as in (3.4) of Lemma 3.5. We state the algorithm in Algorithm 3, and its running time is similar to Algorithm 2. Its correctness is presented in Lemma 3.5.

Corollary 3.1. *Algorithm 3 can be implemented in time $\tilde{O}(k\|X\|_0)$ if it is given oracle access to the distance d, and in time $\tilde{O}(|V| + |E|)$ if the input is an edge-weighted graph $G = (V,E)$ and M is its shortest-path metric.*

Lemma 3.5. *Fix $0 < \epsilon, \delta < \frac{1}{2}$, an integer $k \geq 1$, and a metric space $M(V,d)$. Given a weighted set $X \subseteq V$, and an $(O(1), O(1))$-approximate solution $C^{\mathrm{apx}} \in V^k$ for k-*Median *on X, suppose $\mathcal{F}_{C^{\mathrm{apx}}} = \{f_x : V \to \mathbb{R}_+ \mid x \in X\}$ satisfies $\forall x \in X, c \in V$,*

$$d(x,c) \leq f_x(c) \leq (1+\epsilon) \cdot d(x,c) + \epsilon \cdot d(x, C^{\mathrm{apx}}); \tag{3.4}$$

then Algorithm 3 computes a weighted set $D \subseteq X$ of size

$$O\left(\epsilon^{-2} k \left(k \log k \cdot \mathrm{sdim}_{\max}(\mathcal{F}_{C^{\mathrm{apx}}}) + \log \tfrac{1}{\delta}\right) + k^2 \log \tfrac{1}{\delta}\right),$$

*that with high probability $1-\delta$ is an ϵ-coreset for k-*Median *on X.*

Algorithm 3 Coresets for k-Median on $\mathcal{F}$ with additive distortion

```
compute an $(O(1), O(1))$-approximate solution $C^{\text{apx}}$ for $k$-Median on $X$
for each $x \in X$, let $\sigma_x^{\text{apx}} := w_X(x) \cdot \left(\frac{d(x, C^{\text{apx}})}{\text{cost}(X, C^{\text{apx}})} + \frac{1}{w_X(C^{\text{apx}}(x))}\right)$   ▷ as in Lemma 3.2
for each $x \in X$, let $p_x := \frac{\sigma_x^{\text{apx}}}{\sum_{y \in X} \sigma_y^{\text{apx}}}$
$N := O(\epsilon^{-2} k(k \log k \cdot \text{sdim}_{\max}(\mathcal{F}_{C^{\text{apx}}}) + \log \frac{1}{\delta}) + k^2 \log \frac{1}{\delta})$
draw $N$ independent samples from $X$, each from the distribution $(p_x : x \in X)$   ▷ $\text{sdim}_{\max}$ as in Lemma 3.1, and $\mathcal{F}_{C^{\text{apx}}}$ as in (3.4)
for each $x$ in the sample $D$ assign weight $w_D(x) := \frac{w_X(x)}{p_x N}$
return the weighted set $D$
```

4 Coresets

We now apply the framework developed in Section 3 to design coresets of size independent of X for various settings, including excluded-minor graphs (in Section 4.1), high-dimensional Euclidean spaces (in Section 4.3), and graphs with bounded highway dimension (in Section 4.4). Our workhorse will be Lemma 3.3 and Lemma 3.5, which effectively translate a terminal embedding $\mathcal{F}$ with low distortion on $X \times V$ and low shattering dimension $\text{sdim}_{\max}$ into an efficient algorithm to construct a coreset whose size is linear in $\text{sdim}_{\max}(\mathcal{F})$.

We therefore turn our attention to designing various terminal embeddings. For excluded-minor graphs, we design a terminal embedding $\mathcal{F}$ with multiplicative distortion $1 + \epsilon$ of the distances, and dimension $\text{sdim}_{\max}(\mathcal{F}) = O(\text{poly}(k/\epsilon) \cdot \log \|X\|_0)$. For Euclidean spaces, we employ a known terminal embedding with similar guarantees. In both settings, even though the shattering dimension depends on $\|X\|_0$, it still implies coresets of size independent of X by our iterative size reduction (Theorem 3.1). We thus obtain the first coreset (of size independent of X and V) for excluded-minor graphs (Corollary 4.1), and a simpler state-of-the-art coreset for Euclidean spaces (Corollary 4.2).

We also design a terminal embedding for graphs with bounded highway dimension (formally defined in Section 4.4). This embedding has an additive distortion (on top of the multiplicative one), but its shattering dimension is independent of X, hence the iterative size reduction is not required. We thus obtain the first coreset (of size independent of X and V) for graphs with bounded highway dimension (Corollary 4.3).

4.1 Excluded-minor Graphs Our terminal embedding for excluded-minor graphs is stated in the next lemma. Previously, the shattering dimension of the shortest-path metric of graphs excluding a fixed graph H_0 as a minor was studied only for unit point weight, for which Bousquet and Thomassé [8] proved that $\mathcal{F} = \{d(x, \cdot) \mid x \in X\}$ has shattering dimension $\text{sdim}(\mathcal{F}) = O(|H_0|)$. For arbitrary point weight, i.e., $\text{sdim}_{\max}(\mathcal{F})$, it is still open to get a bound that depends only on $|H_0|$, although the special case of bounded treewidth was recently resolved, as Baker et al. [5], proved that $\text{sdim}_{\max}(\mathcal{F}) = O(\text{tw}(G))$ where $\text{tw}(G)$ denotes the treewidth of the graph G. Note that both of these results use no distortion of the distances, i.e., they bound $\mathcal{F} = \{d(x, \cdot) \mid x \in X\}$. Our terminal embedding handles the most general setting of excluded-minor graphs and arbitrary point weight, although it bypasses the open question by allowing a small distortion and dependence on X.

Lemma 4.1. *For every edge-weighted graph $G = (V, E)$ that excludes some fixed minor and whose shortest-path metric is denoted as $M = (V, d)$, and for every weighted set $X \subseteq V$, there exists a set of functions $\mathcal{F} := \{f_x : V \to \mathbb{R}_+ \mid x \in X\}$ such that*

$$\forall x \in X, c \in V, \qquad d(x, c) \leq f_x(c) \leq (1 + \epsilon) \cdot d(x, c),$$

and $\text{sdim}_{\max}(\mathcal{F}) = \tilde{O}(\epsilon^{-2}) \cdot \log \|X\|_0$.

Let us present now an overview of the proof of Lemma 4.1, deferring the full details to Section 4.2. Our starting point is the following approach, which was developed in [5] for bounded-treewidth graphs. (The main purpose is to explain how vertex separators are used as portals to bound the shattering dimension, but unfortunately additional technical details are needed.) The first step in this approach reduces the task of bounding the shattering dimension to counting how many distinct permutations of X one can obtain by ordering the points of X according to their distance from a point c, when ranging over all $c \in V$. An additional argument uses the bounded treewidth to reduce the range of c from all of V to a subset $\hat{V} \subset V$, that is separated from X by a vertex-cut $P \subset V$ of size $|\hat{P}| = O(1)$. This means that every path, including the shortest-path, between every $x \in X$ and every $c \in \hat{V}$ must pass through $\hat{P}$, therefore

$$d(x, c) = \min\{d(x, p) + d(p, c) : p \in \hat{P}\},$$

and the possible orderings of X are completely determined by these values. The key idea now is to replace the hard-to-control range of $c \in \hat{V}$ with a richer but easier range of $|\hat{P}| = O(1)$ real variables. Indeed, each $d(x, \cdot)$ is captured by a *min-linear function*, which means a function of the form $\min_i a_i y_i + b_i$ with real variables $\{y_i\}$ that represent $\{d(p,c)\}_{p\in\hat{P}}$ and fixed coefficients $\{a_i, b_i\}$. Therefore, each $d(x,\cdot)$ is captured by a min-linear function $g_x : \mathbb{R}^{|\hat{P}|} \to \mathbb{R}_+$, and these functions are all defined on the same $|\hat{P}| = O(1)$ real variables. In this representation, it is easy to handle the point weight $v : X \to \mathbb{R}_+$ (to scale all distances from x), because each resulting function $v(x) \cdot g_x$ is still min-linear. Finally, the number of orderings of the set $\{g_x\}_{x\in X}$ of min-linear functions, is counted using the arrangement number for hyperplanes, which is a well-studied quantity in computational geometry.

To extend this approach to excluded-minor graphs (or even planar graphs), which do not admit small vertex separators, we have to replace vertex separators with shortest-path separators [53, 3]. In particular, we use these separator theorem to partition the whole graph into a few parts, such that each part is separated from the graph by only a few shortest paths, see Lemma 4.3 for planar graphs (which is a variant of a result known from [18]) and Lemma 4.7 for excluded-minor graphs. However, the immediate obstacle is that while these separators consist of a few paths, their total size is unbounded (with respect to X), which breaks the above approach because each min-linear function has too many variables. A standard technique to address this size issue is to discretize the path separator into *portals*, and reroute through them a shortest-path from each $x \in X$ to each $c \in V$. This step distorts the distances, and to keep the distortion bounded multiplicatively by $1+\epsilon$, one usually finds inside each separating shortest-path l, a set of portals $P_l \subset l$ whose spacing is at most $\epsilon \cdot d(x,c)$. However, $d(x,c)$ could be very small compared to the entire path l, hence we cannot control the number of portals (even for one path l).

Vertex-dependent Portals In fact, all we need is to represent the relative ordering of $\{d(x,\cdot) : x \in X\}$ using a set of *min-linear functions* over a few real variables, and these variables do not have to be the distance to *fixed portals* on the separating shortest paths. (Recall this description is eventually used by the arrangement number of hyperplanes to count orderings of X.) To achieve this, we first define *vertex-dependent* portals P_c^l with respect to a separating shortest path l *and* a vertex $c \in V$ (notice this includes also P_x^l for $x \in X$). and then a shortest path from $x \in X$ to $c \in V$ passing through l is rerouted through portals $P_x^l \cup P_c^l$, as follows. First, since l is itself a shortest path, $d(x,c) = \min_{u_1,u_2\in l}\{d(x,u_1) + d(u_1,u_2) + d(u_2,c)\}$. Observe that $d(u_1,u_2)$ is already linear, because one real variable can "capture" a location in l, hence we only need to approximate $d(x,u_1)$ and $d(c,u_2)$. To do so, we approximate the distances from c to every vertex on the path l, i.e., $\{d(c,u)\}_{u\in l}$, using only the distances from c to its portal set P_c^l, i.e., $\{d(c,p)\}_{p\in P_c^l}$. Moreover, between successive portals this approximate distance is a linear function, and it actually suffices to use $|P_c^l| = \text{poly}(1/\epsilon)$ portals, which means that $d(c,u)$ can be represented as a *piece-wise linear* function in $\text{poly}(1/\epsilon)$ real variables.

Note that the above approach ends up with the minimum of piece-wise linear (rather than linear) functions, which creates extra difficulty. In particular, we care about the relative ordering of $\{d(x,\cdot) : x \in X\}$ over all $c \in V$, and to evaluate $d(x,c)$ we need the pieces that c and x generate, i.e., information about $P_c^l \cup P_x^l$. Since the number of $c \in V$ is unbounded, we need to "guess" the structure of P_c^l, specifically the ordering between the portals in P_c^l and those in P_x^l. Fortunately, since every $|P_c^l| \le \text{poly}(1/\epsilon)$, such a "guess" is still affordable, and this would prove Lemma 4.1.

Corollary 4.1. *For every edge-weighted graph $G = (V,E)$ that excludes a fixed minor, every $0 < \epsilon, \delta < 1/2$ and integer $k \ge 1$, k-*Median *of every weighted set $X \subseteq V$ (with respect to the shortest path metric of G) admits an ϵ-coreset of size $\tilde{O}(\epsilon^{-4}k^2 \log \frac{1}{\delta})$. Furthermore, such a coreset can be computed in time $\tilde{O}(|E|)$ with success probability $1-\delta$.*

Remark 4.1. *This result partly extends to (k,z)-*Clustering *for all $z \ge 1$. The importance sampling algorithm and its analysis are immediate, and in particular imply the existence of a coreset of size $\tilde{O}(\epsilon^{-4}k^2 \log \frac{1}{\delta})$. However we rely on known algorithm for $z = 1$ in the step of computing an approximate clustering (needed to compute sampling probabilities).*

4.2 Proof of Lemma 4.1

For the sake of presentation, we start with proving the planar case, since this already requires most of our new technical ideas. The statement of terminal embedding for planar graphs is as follows, and how the proof can be modified to work for the minor-excluded case is briefly discussed in Section 4.2.1.

Lemma 4.2. *For every edge-weighted planar graph $G = (V,E)$ whose shortest path metric is denoted as $M = (V,d)$ and every weighted set $X \subseteq V$, there exists a set of functions $\mathcal{F} = \mathcal{F}_X := \{f_x : V \to \mathbb{R}_+ \mid x \in X\}$ such that for every $x \in X$, and $c \in V$, $f_x(c) \in (1\pm\epsilon)\cdot d(x,c)$, and $\text{sdim}_{\max}(\mathcal{F}) = \tilde{O}(\epsilon^{-2}) \log \|X\|_0$.*

By definition, $\text{sdim}_{\max}(\mathcal{F}) = \max_{v:X\to\mathbb{R}_+}(\mathcal{F}_v)$, so it suffices to bound $\text{sdim}(\mathcal{F}_v)$ for every v. Also, by the definition of sdim, it suffices to prove for every $\mathcal{H} \subseteq \mathcal{F}_v$

with $|\mathcal{H}| \geq 2$,

$$|\{B_{\mathcal{H}}(c,r) : c \in V, r \geq 0\}| \leq \mathrm{poly}(\|X\|_0) \cdot |\mathcal{H}|^{\tilde{O}(\frac{\log \|X\|_0}{\epsilon^2})}$$

Hence, we fix some $v : X \to \mathbb{R}_+$ and $\mathcal{H} \subseteq \mathcal{F}_v$ with $|\mathcal{H}| \geq 2$ throughout the proof.

General Reduction: Counting Relative Orderings For $\mathcal{H} \subseteq \mathcal{F}$ and $c \in V$, let $\sigma_c^{\mathcal{H}}$ be the permutation of $\mathcal{H}$ ordered by $v(x) \cdot f_x(c)$ in non-decreasing order and ties are broken arbitrarily. Then for a fixed $c \in V$ and very $r \geq 0$, the subset $B_{\mathcal{H}}(c,r) \subseteq \mathcal{H}$ is exactly the subset defined by some prefix of $\sigma_c^{\mathcal{H}}$. Hence,

$$|\{B_{\mathcal{H}}(c,r) : c \in V, r \geq 0\}| \leq |\mathcal{H}| \cdot \left|\{\sigma_c^{\mathcal{H}} : c \in V\}\right|.$$

Therefore, it suffices to show

$$\left|\{\sigma_c^{\mathcal{H}} : c \in V\}\right| \leq \mathrm{poly}(\|X\|_0) |\mathcal{H}|^{\tilde{O}(\epsilon^{-2}) \log \|X\|_0}$$

Hence, this reduces the task of bounding of shattering dimension to counting the number of relative orderings of $\{v(x) \cdot f_x(c) \mid x \in X\}$.

Next, we use the following structural lemma for planar graphs to break the graph into few parts of simple structure, so we can bound the number of permutations for c coming from each part. A variant of this lemma has been proved in [18], where the key idea is to use the *interdigitating* trees.

Lemma 4.3. (see also [18]) *For every edge-weighted planar graph $G = (V,E)$ and subset $S \subseteq V$, V can be broken into parts $\Pi := \{V_i\}_i$ with $|\Pi| = \mathrm{poly}(|S|)$ and $\bigcup_i V_i = V$, such that for every $V_i \in \Pi$,*

1. *$|S \cap V_i| = O(1)$,*
2. *there exists a collection of shortest paths $\mathcal{P}_i$ in G with $|\mathcal{P}_i| = O(1)$ and removing the vertices of all paths in $\mathcal{P}_i$ disconnects V_i from $V \setminus V_i$ (points in V_i are possibly removed).*

Applying Lemma 4.3 with $S = X$ (noting that S is an unweighted set), we obtain $\Pi = \{V_i\}_i$ with $|\Pi| = \mathrm{poly}(\|X\|_0)$, such that each part $V_i \in \Pi$ is separated by $O(1)$ shortest paths $\mathcal{P}_i$. Then

$$\left|\{\sigma_c^{\mathcal{H}} : c \in V\}\right| \leq \sum_{V_i \in \Pi} \left|\{\sigma_c^{\mathcal{H}} : c \in V_i\}\right|.$$

Hence it suffices to show for every $V_i \in \Pi$, it holds that

$$\left|\{\sigma_c^{\mathcal{H}} : c \in V_i\}\right| \leq |\mathcal{H}|^{\tilde{O}(\epsilon^{-2}) \log \|X\|_0}. \tag{4.5}$$

Since $\bigcup_i V_i = V$, it suffices to define functions $f_x(\cdot)$ for $c \in V_i$ for every i independently. Therefore, we fix $V_i \in \Pi$ throughout the proof. In the following, our proof proceeds in three parts. The first defines functions $f_x(\cdot)$ on V_i, the second analyzes the distortion of f_x's, and the final part analyzes the shattering dimension.

Part I: Definition of f_x on V_i By Lemma 4.3 we know $|V_i \cap X| = O(1)$. Hence, the "simple" case is when $x \in V_i \cap T$, for which we define $f_x(\cdot) := d(x, \cdot)$.

Otherwise, $x \in X \setminus V_i$. Write $\mathcal{P}_i := \{P_j\}_j$. Since P_j's are shortest paths in G, and removing $\mathcal{P}_i$ from G disconnects V_i from $V \setminus V_i$, we have the following fact.

Fact 4.1. *For $c \in V_i$ and $x \in X \setminus V_i$, there exists $P_j \in \mathcal{P}_i$ and $c', x' \in P_j$, such that $d(c,x) = d(c,c') + d(c',x') + d(x',x)$.*

Let $d_j(c,x)$ be the length of the shortest path from c to x that uses *at least one* point in P_j. For each $P_j \in \mathcal{P}_i$, we will define $f_x^j : V_i \to \mathbb{R}_+$, such that $f_x^j(c)$ is within $(1 \pm \epsilon) \cdot d_j(c,x)$, and let

$$f_x(c) := \min_{P_j \in \mathcal{P}_i} f_x^j(c), \qquad \forall c \in V_i.$$

Hence, by Fact 4.1, the guarantee that $f_x^j(c) \in (1 \pm \epsilon) \cdot d_j(c,x)$ implies $f_x(c) \in (1 \pm \epsilon) \cdot d(x,c)$, as desired. Hence we focus on defining f_x^j in the following.

Defining $f_x^j : V_i \to \mathbb{R}_+$ Suppose we fix some $P_j \in \mathcal{P}_i$, and we will define $f_x^j(c)$, for $c \in V_i$. By Fact 4.1 and the optimality of shortest paths, we have

$$d_j(x,c) = \min_{c',x' \in P_j} \{d(c,c') + d(c',x') + d(x',x)\}.$$

For every $y \in V$, we will define $l_y^j : P_j \to \mathbb{R}_+$ such that $l_y^j(y') \in (1 \pm \epsilon) \cdot d(y,y')$ for every $y' \in P_j$. Then, we let

$$f_x^j(c) := \min_{c',x' \in P_j} \{l_c^j(c') + d(c',x') + l_x^j(x')\},$$

and this would imply $f_x^j(c) \in (1 \pm \epsilon) \cdot d_j(x,c)$. So it remains to define $l_y^j : P_j \to \mathbb{R}_+$ for every $y \in V$.

Defining $l_y^j : P_j \to \mathbb{R}_+$ Fix $y \in V$ and we will define $l_y^j(y')$ for every $y' \in P_j$. Pick $h_y \in P_j$ that satisfies $d(y, h_y) = d(y, P_j)$. Since P_j is a shortest path, we interpret P_j as a segment in the real line. In particular, we let the two end points of P_j be 0 and 1, and P_j is a (discrete) subset of $[0,1]$.

Define $a, b \in P_j$ such that $a \leq h_y \leq b$ are the two *furthest* points on the two sides of h on P_j that satisfy $d(h_y, a) \leq \frac{d(y,h_y)}{\epsilon}$ and $d(h_y, b) \leq \frac{d(y,h_y)}{\epsilon}$. Then construct a sequence of points $a = q_1 \leq q_2 \ldots$ in the following way. For $t = 1, 2, \ldots$, if there exists $u \in (q_t, 1] \cap P_j$ such that $d(q_t, u) > \epsilon \cdot d(y, h_y)$, then let q_{t+1} be the smallest such u; if such u does not exist, then let $q_{t+1} := b$ and terminate. Essentially, this breaks P_j into segments of length $\epsilon \cdot d(y, h_y)$, except that the last one that ends with b may be shorter. Denote this sequence as $Q_y := (q_1 = a, \ldots, q_m = b)$.

Claim 4.1. *For every $y \in V$, $|Q_y| = O(\epsilon^{-2})$.*

Proof. By the definition of Q_y, for $1 \leq t \leq m-2$, $d(q_t, q_{t+1}) > \epsilon \cdot d(y, h_y)$. On the other hand, by the definition of a and b, $d(q_1, q_m) = d(a,b) \leq O(\frac{d(y,h_y)}{\epsilon})$. Therefore, $|Q_y| \leq O(\epsilon^{-2})$, as desired. □

Definition of f_x on V_i: Recap Define

(4.6)
$$l_y^j(y') := \begin{cases} d(h_y, y') & \text{if } y' < a = q_1 \text{ or } y' > b = q_m \\ d(y, q_t) & \text{if } q_t \leq y' < q_{t+1}, 1 \leq t < m \\ d(y, q_m) & \text{if } y' = b = q_m \end{cases}$$

where $h_y \in P_j$, $Q_y = \{q_t\}_t \subset P_j$. To recap,

- if $x \in X \cap V_i$, then $f_x(c) := d(x,c)$;
- otherwise $x \in X \setminus V_i$, $f_x(c) := \min_{P_j \in \mathcal{P}_i} f_x^j(c)$, where

$$(4.7)\quad f_x^j(c) := \min_{c', x' \in P_j} \{l_c^j(c') + d(c', x') + l_x^j(x')\}.$$

Finally,

$$(4.8)\qquad f_x(c) := \min_{P_j \in \mathcal{P}_i} f_x^j(c), \qquad \forall c \in V_i.$$

Part II: Distortion Analysis The distortion of l's is analyzed in the following Lemma 4.4, and the distortion for f_x follows immediately from the above definitions.

Lemma 4.4. *For every $P_j \in \mathcal{P}_i$, $y \in V$, $y' \in P_j$, $l_y^j(y') \in (1 \pm \epsilon) \cdot d(y, y')$.*

Proof. If $y' = q_m = b$, by definition $l_y^j(y') = d(y, q_m) = d(y, y')$. Then consider the case when $y' < a = q_1$ or $y' > b = q_m$.

$$\begin{aligned} l_y^j(y') &= d(h_y, y') \\ &\in d(y', y) \pm d(y, h_y) \\ &\in d(y', y) \pm \epsilon \cdot d(y', h_y), \end{aligned}$$

where the last inequality follows from $d(y', h_y) > \frac{d(y,h_y)}{\epsilon}$. This implies $d(y, y') \in (1 \pm \epsilon) \cdot l_y^j(y')$.

Otherwise, $q_t \leq y' < q_{t+1}$ for some $1 \leq t < m$. By the definition of q_t's and the definition of h_y,

$$\begin{aligned} d(y, y') &\in d(y, q_t) \pm d(q_t, y') \\ &\in d(y, q_t) \pm \epsilon \cdot d(y, h_y) \\ &\in d(y, q_t) \pm \epsilon \cdot d(y, y') \\ &\in l_y^j(y') \pm \epsilon \cdot d(y, y'), \end{aligned}$$

which implies $l_y^j(y') \in (1 \pm \epsilon) \cdot d(y, y')$. This finishes the proof of Lemma 4.4. □

Part III: Shattering Dimension Analysis Recall that we fixed $v : X \to \mathbb{R}_+$ and $\mathcal{H} \subseteq \mathcal{F}_v$ with $|\mathcal{H}| \geq 2$. Now we show

$$(4.9)\qquad \left|\{\sigma_c^{\mathcal{H}} : c \in V_i\}\right| \leq |\mathcal{H}|^{\tilde{O}(\epsilon^{-2}) \log \|X\|_0}.$$

Let $H := \{x : v(x) \cdot f_x \in \mathcal{H}\}$, so $|H| = |\mathcal{H}|$. Recall that $|V_i \cap X| = O(1)$ by Lemma 4.3, so $|V_i \cap H| = O(1)$. Hence, if we could show

$$\left|\{\sigma_c^{\mathcal{H}} : c \in V_i\}\right| \leq N(|H|)$$

for $\mathcal{H}$ such that $H \cap V_i = \emptyset$, then for general $\mathcal{H}$,

$$\begin{aligned} \left|\{\sigma_c^{\mathcal{H}} : c \in V_i\}\right| &\leq N(|H| - |V_i \cap H|) \cdot |H|^{O(|V_i \cap H|)} \\ &\leq N(|H|) \cdot |H|^{O(1)} \end{aligned}$$

Therefore, it suffices to show (4.9) under the assumption that $H \cap V_i = \emptyset$.

In the following, we will further break V_i into $|H|^{\tilde{O}(\epsilon^{-2})}$ parts, such that for each part V', f_x on V' may be alternatively represented as a *min-linear* function.

Lemma 4.5. *Let $u = |\mathcal{P}_i|$. There exists a partition Γ of V_i, such that the following holds.*

1. *$|\Gamma| \leq |H|^{\tilde{O}(\epsilon^{-2}) \cdot u}$.*
2. *$\forall V' \in \Gamma$, $\forall x \in H$, there exists $g_x : \mathbb{R}^s \to \mathbb{R}_+$ where $s = O(\epsilon^{-2})$, such that g_x is a minimum of $O(\epsilon^{-4} u)$ linear functions on $\mathbb{R}^s$, and for every $c \in V'$, there exists $y \in \mathbb{R}^s$ that satisfies $f_x(c) = g_x(y)$.*

Proof. Before we actually prove the lemma, we need to examine $f_x^j(c)$ and l_y^j more closely. Suppose some $P_j \in \mathcal{P}_i$ is fixed. Recall that for $y \in V, y' \in P_j$ (defined in (4.6)),

$$l_y^j(y') := \begin{cases} d(h_y, y') & \text{if } y' < a = q_1 \text{ or } y' > b = q_m \\ d(y, q_t) & \text{if } q_t \leq y' < q_{t+1}, 1 \leq t < m \\ d(y, q_m) & \text{if } y' = b = q_m \end{cases}$$

where $h_y \in P_j$, $Q_y = \{q_t\}_t \subset P_j$. Hence, for every y, l_y^j is a *piece-wise linear* function with $O(|Q_y|) = O(\epsilon^{-2})$ (by Claim 4.1) pieces, where the transition points of l_y^j are $Q_y \cup \{0, 1\}$ (noting that $d(h_y, y')$ is linear since $h_y, y' \in P_j$).

Using that l's are piece-wise linear, we know for $c \in V_i, x \in X \setminus V_i$,

$$\begin{aligned} f_x^j(c) &= \min_{c', x' \in P_j} \{l_c^j(c') + d(c', x') + l_x^j(x')\} \\ &= \min_{c', x' \in Q_c \cup Q_x \cup \{0,1\}} \{l_c^j(c') + d(c', x') + l_x^j(x')\}. \end{aligned}$$

where the first equality is by definition in (4.7) and the second equality is because l's are piece-wise linear.

Hence, to evaluate $f_x^j(c)$ we only need to evaluate $l_c^j(c')$ and $l_x^j(x')$ at $c', x' \in Q_c \cup Q_x \cup \{0,1\}$, and in particular we need to find the piece in l_c^j and l_x^j that every $c', x' \in Q_c \cup Q_x \cup \{0,1\}$ belong to, and then evaluate a linear function. Precisely, the piece that every c', x' belongs to is determined by the relative ordering of points $Q_x \cup Q_c$ (recalling that they are from P_j). Thus, the pieces are not only determined by x, but also by c which is the variable, and this means without the information about the pieces, f_x cannot be represented as a min-linear function g_x. Therefore, the idea is to find a partition Γ of V_i, such that for c in each part $V' \in \Gamma$, the relative ordering of Q_c with respect to $\{Q_x : x \in H\}$ is the same. We note that we need to consider the ordering of Q_c with respect to all Q_x's, because we care about the relative orderings of all f_x's.

Defining Γ For $1 \le j \le u$, $c \in V_i$, let τ_c^j be the ordering of Q_c with respect to $\bigcup_{y \in H} Q_y$ on P_j. Here, an ordering of Q_c with respect to $\left(\bigcup_{y \in H} Q_y\right)$ is defined by their ordering on P_j which is interpreted as the real line. In our definition of Γ, we will require each part $V' \in \Gamma$ to satisfy that $\forall c \in V'$, the tuple of orderings $(\tau_c^1, \ldots, \tau_c^u)$ remains the same. That is, V_i is partitioned according to the joint relative ordering τ_c^j's on all shortest paths $P_j \in \mathcal{P}_i$.

Formally, for $1 \le j \le u$, let $\Lambda^j := \{\tau_c^j : c \in V_i\}$ be the collection of distinct ordering τ_c^j on P_j over points $c \in V_i$. Define

$$\Lambda := \Lambda^1 \times \ldots \times \Lambda^u$$

as the tuples of τ_j's for $1 \le j \le u$ (here, the $\times$ operator is the Cartesian product). For $(\tau_1, \ldots, \tau_u) \in \Lambda$, define

$$V_i^{(\tau_1, \ldots, \tau_u)} := \{c \in V_i : (\tau_c^1 = \tau_1) \wedge \ldots \wedge (\tau_c^u = \tau_u)\}$$

as the subset of V_i such that the ordering τ_c^j for each $1 \le j \le u$ agrees with the given tuple. Finally, we define the partition as

$$\Gamma := \{V_i^{(\tau_1, \ldots, \tau_u)} : (\tau_1, \ldots, \tau_u) \in \Lambda\}.$$

Bounding $|\Gamma|$ By Claim 4.1, we know $|Q_y| = O(\epsilon^{-2})$ for every $y \in V$. Hence, $\left|\bigcup_{y \in H} Q_y\right| = O\left(\epsilon^{-2}|H|\right)$. Therefore, for every $j \in [u]$,

$$|\Lambda^j| \le \binom{O(\epsilon^{-2}|H|)}{O(\epsilon^{-2})} = O\left(\epsilon^{-1}|H|\right)^{O(\epsilon^{-2})}.$$

Therefore,

$$|\Gamma| \le \Pi_{1 \le j \le u} |\Lambda^j| \le O\left(\epsilon^{-1}|H|\right)^{O(\epsilon^{-2}u)} \le |H|^{\tilde{O}(\epsilon^{-2}) \cdot u},$$

as desired.

Defining g_x By our definition of Γ, we need to define g_x for each $V' \in \Gamma$. Now, fix tuple $(\tau_1, \ldots, \tau_u) \in \Lambda$, so the part corresponds to this tuple is $V' = V_i^{(\tau_1, \ldots, \tau_u)}$, and we will define g_x with respect to such V'. Similar to the definition of f_x's (see (4.8)), we define $g_x : \mathbb{R}^s \to \mathbb{R}_+$ to have the form

$$g_x(y) := \min_{P_j \in \mathcal{P}_i} g_x^j(y).$$

Then, for $1 \le j \le u$, $x \in H$, define $g_x^j : \mathbb{R}^s \to \mathbb{R}$ of $s := O(\epsilon^{-2})$ variables $(q_1, \ldots, q_m, d(c, q_1), \ldots, d(c, q_m), h_c)$ for $q_i \in Q_c$, such that

$$\begin{aligned} & g_x^j(q_1, \ldots, q_m, d(c, q_1), \ldots, d(c, q_m), h_c) \\ = & \min_{c', x' \in Q_c \cup Q_x \cup \{0,1\}} \{l_c^j(c') + d(c', x') + l_x^j(x')\}. \end{aligned}$$

We argue that for every $1 \le j \le u$, g_x^j may be viewed as a minimum of $O(\epsilon^{-4})$ linear functions whose variables are the same with that of g_x^j.

- Linearity. Suppose $c \in V'$, and fix $c', x' \in Q_c \cup Q_x \cup \{0,1\}$. By the above discussions, $l_c^j(c')$ could take values only from $\{d(c, q_i) : q_i \in Q_c\} \cup \{d(h_c, c')\}$. Since $\forall q_i \in Q_c$, $d(c, q_i)$ is a variable of g_x^j, and $d(h_c, c') = |h_c - c'|$ is linear and that h_c is also a variable of g_x^j, we conclude that $l_c^j(c')$ may be written as a linear function of the same set of variables of g_x^j. By a similar argument, we have the same conclusion for l_x^j. Therefore, $l_c^j(c') + d(c', x') + l_x^j(x')$ may be written as a linear function of $(q_1, \ldots, q_m, d(c, q_1), \ldots, d(c, q_m), h_c)$.
- Number of linear functions. By Claim 4.1, we have
$$\forall y \in V, \qquad |Q_y| = O(\epsilon^{-2}),$$
hence $|Q_c \cup Q_x \cup \{0,1\}| = O(\epsilon^{-2})$. Therefore, there are $O(\epsilon^{-4})$ pairs of $c', x' \in Q_c \cup Q_x \cup \{0,1\}$.

Therefore, item 2 of Lemma 4.5 follows by combining this with the definition of g_x. We completed the proof of Lemma 4.5. □

Now suppose Γ is the one that is guaranteed by Lemma 4.5. Since

$$\left|\{\sigma_c^{\mathcal{H}} : c \in V_i\}\right| \le \sum_{V' \in \Gamma} \left|\{\sigma_c^{\mathcal{H}} : c \in V'\}\right|$$

and

$$|\Gamma| \le |H|^{\tilde{O}(\epsilon^{-2}) \cdot u} \le |H|^{\tilde{O}(\epsilon^{-2})}, \tag{4.10}$$

where the last inequality is by Lemma 4.3 (recalling $u = |\mathcal{P}_i|$), it suffices to show for every $V' \in \Gamma$,

$$\left|\{\sigma_c^{\mathcal{H}} : c \in V'\}\right| \le |H|^{\tilde{O}(\epsilon^{-2}) \log \|X\|_0}. \tag{4.11}$$

Fix some $V' \in \Gamma$. By Lemma 4.5, for every $x \in H$ there exists a min-linear function $g_x : \mathbb{R}^s \to \mathbb{R}_+$ ($s = O(\epsilon^{-2})$)), such that for every $c \in V'$, there exists $y \in \mathbb{R}^s$ that satisfies $f_x(c) = g_x(y)$. For $y \in \mathbb{R}^s$ define π^H_y as a permutation of H that is ordered by $g_x(y)$ in non-increasing order and ties are broken in a way that is consistent with σ. Then

$$\left|\{\sigma^{\mathcal{H}_v}_c : c \in V'\}\right| \le \left|\{\pi^H_y : y \in \mathbb{R}^s\}\right|. \tag{4.12}$$

We make use of the following lemma to bound the number of permutations π^H_y. The lemma relates the number of relative orderings of g_x's to the arrangement number in computational geometry.

Lemma 4.6. ([5]) *Suppose there are m functions $g_1, \ldots, g_m$ from $\mathbb{R}^s$ to $\mathbb{R}$, such that $\forall i \in [m]$, g_i is of the form*

$$g_i(x) := \min_{j\in[t]}\{g_{ij}(x)\},$$

where g_{ij} is a linear function. For $x \in \mathbb{R}^s$, let π_x be the permutation of $[m]$ ordered by $g_i(x)$. Then,

$$|\{\pi_x : x \in \mathbb{R}^s\}| \le (mt)^{O(s)}.$$

Applying Lemma 4.6 on g_x's for $x \in H$ with parameters $s = O(\epsilon^{-2})$, $t = O(\epsilon^{-4}u) = O\left(\epsilon^{-4}\log\|X\|_0\right)$ and $m = |H|$, we obtain

$$\left|\{\pi^H_y : y \in \mathbb{R}^s\}\right| \le |H|^{\tilde{O}(\epsilon^{-2})\cdot\log\|X\|_0}. \tag{4.13}$$

Thus, (4.11) is implied by combining (4.13) with (4.12). Finally, we complete the proof of Lemma 4.2 by combining the above three parts of the arguments.

4.2.1 From Planar to Minor-excluded Graphs

The strategy for proving the minor-excluded case is similar to the planar case. Due to the space limit, we only present the structural lemma (Lemma 4.7) that replaces Lemma 4.3 which we used for planar graphs, and highlight the differences.

Lemma 4.7. *Given edge-weighted graph $G = (V, E)$ that excludes a fixed minor, and a subset $S \subseteq V$, there is a collection $\Pi := \{V_i\}_i$ of V with $|\Pi| = \mathrm{poly}(|S|)$ and $\bigcup_i V_i = V$ such that for every $V_i \in \Pi$ the following holds.*

1. *$|S \cap V_i| = O(1)$.*
2. *There exists an integer t_i and t_i groups of paths $\mathcal{P}^i_1, \ldots, \mathcal{P}^i_{t_i}$ in G, such that*
   - *(a) $|\bigcup_{j=1}^{t_i}\mathcal{P}^i_j| = O(\log|S|)$*
   - *(b) removing the vertices of all paths in $\bigcup_{j=1}^{t_i}\mathcal{P}^i_j$ disconnects V_i from $V \setminus V_i$ in G (possibly removing points in V_i)*
   - *(c) for $1 \le j \le t_i$, let G^i_j be the sub-graph of G formed by removing all paths in $\mathcal{P}^i_1, \ldots, \mathcal{P}^i_{j-1}$ (define $G^i_1 = G$), then every path in $\mathcal{P}^i_j$ is a shortest path in G^i_j.*

The lemma follows from a recursive application of the balanced shortest path separator theorem in [3, Theorem 1]. Compared with Lemma 4.3, the separating shortest paths in Lemma 4.7 are not from the original graph G, but is inside some sub-graph generated by removing various other separating shortest paths. Also, the number of shortest paths in the separator is increased from $O(1)$ to $O(\log\|X\|_0)$. The remaining proof for the excluded-minor case can be found in the full version.

4.3 High-Dimensional Euclidean Spaces

We present a terminal embedding for Euclidean spaces, with a guarantee that is similar to that of excluded-minor graphs. For these results, the ambient metric space (V, d) of all possible centers is replaced by a Euclidean space.[8]

Lemma 4.8. *For every $\epsilon \in (0, 1/2)$ and finite weighted set $X \subset \mathbb{R}^m$, there exists $\mathcal{F} = \{f_x : \mathbb{R}^m \to \mathbb{R}_+ \mid x \in X\}$ such that*

$$\forall x \in X, c \in \mathbb{R}^m, \qquad \|x - c\|_2 \le f_x(c) \le (1+\epsilon)\|x - c\|_2,$$

and $\mathrm{sdim}_{\max}(\mathcal{F}) = O(\epsilon^{-2}\log\|X\|_0)$.

Proof. The lemma follows immediately from the following terminal version of the Johnson-Lindenstrauss Lemma [36], proved recently by Narayanan and Nelson [46].

Theorem 4.1. ([46]) *For every $\epsilon \in (0, 1/2)$ and finite $S \subset \mathbb{R}^m$, there is an embedding $g : S \to \mathbb{R}^t$ for $t = O(\epsilon^{-2}\log|S|)$, such that $\forall x \in S, y \in \mathbb{R}^m$,*

$$\|x - y\|_2 \le \|g(x) - g(y)\|_2 \le (1+\epsilon)\|x - y\|_2.$$

Given $X \subset \mathbb{R}^m$, apply Theorem 4.1 with $S = X$ (as an unweighted set), and define for every $x \in X$ the function $f_x(c) := \|g(x) - g(c)\|_2$. Then $\mathcal{F} = \{f_x \mid x \in X\}$ clearly satisfies the distortion bound. The dimension bound follows by plugging $t = O(\epsilon^{-2}\log\|X\|_0)$ into

[8] It is easily verified that as long as X is finite, our entire framework from Section 3 extends to $V = \mathbb{R}^m$ with ℓ_2 norm. For example, all maximums (e.g., in Lemma 3.1) are well-defined by using compactness arguments on a bounding box.

the bound $\mathrm{sdim}_{\max}(\mathcal{F}) = O(t)$ known from [20, Lemma 16.3].[9] □

Corollary 4.2. *For every $0 < \epsilon, \delta < 1/2$, $z \geq 1$, and integers $k, m \geq 1$, Euclidean (k,z)-Clustering of every weighted set $X \subset \mathbb{R}^m$ admits an ϵ-coreset of size $\tilde{O}(\epsilon^{-4} 2^{2z} k^2 \log \frac{1}{\delta})$. Furthermore, such a coreset can be computed[10] in time $\tilde{O}(k\|X\|_0 m)$ with success probability $1-\delta$.*

Remark 4.2. (Comparison to [33]) *For (k,z)-Clustering in Euclidean spaces, our algorithms can also compute an ϵ-coreset of size $\tilde{O}(\epsilon^{-O(z)}k)$, which offers a different parameters tradeoff than Corollary 4.2. This alternative bound is obtained by simply replacing the application of Lemma 3.1 (which is actually from [22]) with [33, Lemma 3.1] (which itself is a result from [20], extended to weighted inputs).*

Our two coreset size bounds are identical to the state-of-the-art bounds proved by Huang and Vishnoi [33] (in the asymptotic sense). Their analysis is different, and bounds $\mathrm{sdim}_{\max}$ independently of X using a dimensionality-reduction argument for clustering objectives. In contrast, we require only a loose bound $\mathrm{sdim}_{\max}(\mathcal{F}) = O(\mathrm{poly}(\epsilon^{-1}) \cdot \log \|X\|_0)$, which follows immediately from [46], and the coreset size is then reduced iteratively using Theorem 3.1, which simplifies the analysis greatly.

4.4 Graphs with Bounded Highway Dimension

The notion of highway dimension was proposed by Abraham, Fiat, Goldberg, and Werneck [2] to measure the complexity of road networks. Motivated by the empirical observation that a shortest path between two far-away cities always passes through a small number of hub cities, the highway dimension is defined, roughly speaking, as the maximum size of a hub set that meets every long shortest path, where the maximum is over all localities of all distance scale. Several slightly different definitions of highway dimension appear in the literature, and we use the one proposed in [23].

Definition 4.1. (Highway Dimension [23]) *Fix some universal constant $\rho \geq 4$. The highway dimension of an edge-weighted graph $G = (V,E)$, denoted $\mathrm{hdim}(G)$, is the smallest integer t such that for every $r \geq 0$ and $x \in V$, there is a subset $S \subseteq B(x, \rho r)$ with $|S| \leq t$, such that S intersects every shortest path of length at least r all of whose vertices lie in $B(x, \rho r)$.*

Remark 4.3. *This version generalizes the original one from [2] (and also the subsequent journal version [1]), and it was shown to capture a broader range of real-world transportation networks [23]. We also note that the version in [1] is stronger than the notion of doubling dimension [26], however, the version that we use (from [23]) is not. In particular, it means that the previous coreset result for doubling metrics [31] does not apply to our case.*

Unlike the excluded-minor and Euclidean cases mentioned in earlier sections, our coresets for graphs with bounded highway dimension are obtained using terminal embeddings with an additive distortion.

Lemma 4.9. *Let $G = (V,E)$ be an edge-weighted graph and denote its shortest-path metric by $M(V,d)$. Then for every $0 < \epsilon < 1/2$, weighted set $X \subseteq V$ and an (unweighted) subset $S \subseteq V$, there exists $\mathcal{F}_S = \{f_x : V \to \mathbb{R}_+ \mid x \in X\}$ such that $\forall x \in X, c \in V$,*

$$d(x,c) \leq f_x(c) \leq (1+\epsilon) \cdot d(x,c) + \epsilon \cdot d(x,S),$$

and $\mathrm{sdim}_{\max}(\mathcal{F}_S) = (|S| + \mathrm{hdim}(G))^{O(\log(1/\epsilon))}$.

Proof. We rely on an embedding of graphs with bounded highway dimension into graphs with bounded treewidth, as follows.

Lemma 4.10. ([7]) *For every $0 < \epsilon < 1/2$, edge-weighted graph $G = (V,E)$ of highway dimension h, and $S \subseteq V$, there exists a graph $G' = (V', E')$ of treewidth $\mathrm{tw}(G') = (|S| + h)^{O(\log(1/\epsilon))}$, and a mapping $\phi : V \to V'$ such that $\forall x, y \in V$,*

$$\begin{aligned} & d_G(x,y) \\ \leq\ & d_{G'}(\phi(x), \phi(y)) \\ \leq\ & (1+\epsilon) \cdot d_G(x,y) + \epsilon \cdot \min\{d(x,S), d(y,S)\}. \end{aligned}$$

We now apply on G' (the graph produced by Lemma 4.10), the following result from [5, Lemma 3.5], which produces the function set $\mathcal{F}_S$ we need for our proof.

Lemma 4.11. ([5]) *Let $G = (V,E)$ be an edge-weighted graph, and denote its shortest-path metric by $M(V,d)$. Then for every weighted set $X \subseteq V$, the function set $\mathcal{F} = \{d(x,\cdot) \mid x \in X\}$ has $\mathrm{sdim}_{\max}(\mathcal{F}) = O(\mathrm{tw}(G))$, where $\mathrm{tw}(G)$ is the treewidth of G.*

Notice that we could also apply on G' our own Lemma 4.1, because bounded-treewidth graphs are also excluded-minor graphs, however Lemma 4.11 has better dependence on $\mathrm{tw}(G)$ and also saves a $\mathrm{poly}(1/\epsilon)$ factor. This concludes the proof of Lemma 4.9. □

[9] The following is proved in [20, Lemma 16.3]. For every $S \subset \mathbb{R}^t$, the function set $\mathcal{H} := \{h_x \mid x \in S\}$ given by $h_x(y) = \|x-y\|_2$, has shattering dimension $\mathrm{sdim}_{\max}(\mathcal{H}) = O(t)$.

[10] We assume that evaluating $\|x-y\|_2$ for $x, y \in \mathbb{R}^m$ takes time $O(m)$.

Corollary 4.3. *For every edge-weighted graph $G = (V, E)$, $0 < \epsilon, \delta < 1/2$, and integer $k \geq 1$, k-*Median *of every weighted set $X \subseteq V$ (with respect to the shortest path metric of G) admits an ϵ-coreset of size $\tilde{O}((k + \mathrm{hdim}(G))^{O(\log(1/\epsilon))}) \log \frac{1}{\delta})$. Furthermore, it can be computed in time $\tilde{O}(|E|)$ with success probability $1-\delta$.*

References

[1] I. Abraham, D. Delling, A. Fiat, A. V. Goldberg, and R. F. Werneck. Highway dimension and provably efficient shortest path algorithms. *J. ACM*, 63(5):41:1–41:26, 2016.

[2] I. Abraham, A. Fiat, A. V. Goldberg, and R. F. F. Werneck. Highway dimension, shortest paths, and provably efficient algorithms. In *SODA*, pages 782–793. SIAM, 2010.

[3] I. Abraham and C. Gavoille. Object location using path separators. In *PODC*, pages 188–197. ACM, 2006.

[4] S. Arora, P. Raghavan, and S. Rao. Approximation schemes for Euclidean k-medians and related problems. In *Proceedings of the 30th Annual ACM Symposium on Theory of computing*, pages 106–113, 1998.

[5] D. Baker, V. Braverman, L. Huang, S. H. Jiang, R. Krauthgamer, and X. Wu. Coresets for clustering in graphs of bounded treewidth. In *ICML*, Proceedings of Machine Learning Research, 2020. To appear.

[6] L. Becchetti, M. Bury, V. Cohen-Addad, F. Grandoni, and C. Schwiegelshohn. Oblivious dimension reduction for k-means: beyond subspaces and the Johnson-Lindenstrauss lemma. In *Proceedings of the 51st Annual Symposium on Theory of Computing*, pages 1039–1050, 2019.

[7] A. Becker, P. N. Klein, and D. Saulpic. Polynomial-time approximation schemes for k-center, k-median, and capacitated vehicle routing in bounded highway dimension. In *ESA*, volume 112 of *LIPIcs*, pages 8:1–8:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. `https://arxiv.org/abs/1707.08270`.

[8] N. Bousquet and S. Thomassé. VC-dimension and Erdős–Pósa property. *Discret. Math.*, 338(12):2302–2317, 2015.

[9] V. Braverman, D. Feldman, and H. Lang. New frameworks for offline and streaming coreset constructions. *CoRR*, abs/1612.00889, 2016.

[10] V. Braverman, S. H. Jiang, R. Krauthgamer, and X. Wu. Coresets for ordered weighted clustering. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 744–753. PMLR, 2019.

[11] J. Byrka, T. Pensyl, B. Rybicki, A. Srinivasan, and K. Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 737–756. SIAM, 2014.

[12] K. Chen. On coresets for k-Median and k-Means clustering in metric and Euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009.

[13] K. L. Clarkson and D. P. Woodruff. Sketching for M-estimators: A unified approach to robust regression. In *SODA*, pages 921–939. SIAM, 2015.

[14] M. B. Cohen, S. Elder, C. Musco, C. Musco, and M. Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*, pages 163–172, 2015.

[15] V. Cohen-Addad, P. N. Klein, and C. Mathieu. Local search yields approximation schemes for k-means and k-median in Euclidean and minor-free metrics. *SIAM Journal on Computing*, 48(2):644–667, 2019.

[16] V. Cohen-Addad, M. Pilipczuk, and M. Pilipczuk. Efficient approximation schemes for uniform-cost clustering problems in planar graphs. In *ESA*, volume 144 of *LIPIcs*, pages 33:1–33:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[17] W. Cui, H. Zhou, H. Qu, P. C. Wong, and X. Li. Geometry-based edge clustering for graph visualization. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1277–1284, 2008.

[18] D. Eisenstat, P. N. Klein, and C. Mathieu. Approximating k-center in planar graphs. In *SODA*, pages 617–627. SIAM, 2014.

[19] D. Feldman, A. Fiat, and M. Sharir. Coresets for weighted facilities and their applications. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '06, page 315–324. IEEE Computer Society, 2006.

[20] D. Feldman and M. Langberg. A unified framework for approximating and clustering data. In *STOC*, pages 569–578. ACM, 2011. `https://arxiv.org/abs/1106.1379`.

[21] D. Feldman, M. Monemizadeh, C. Sohler, and D. P. Woodruff. Coresets and sketches for high dimensional subspace approximation problems. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, page 630–649. SIAM, 2010.

[22] D. Feldman, M. Schmidt, and C. Sohler. Turning big data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering. *SIAM Journal on Computing*, 49(3):601–657, 2020.

[23] A. E. Feldmann, W. S. Fung, J. Könemann, and I. Post. A $(1+\varepsilon)$-embedding of low highway dimension graphs into bounded treewidth graphs. *SIAM Journal on Computing*, 47(4):1667–1704, 2018.

[24] Z. Feng, P. Kacham, and D. P. Woodruff. Strong coresets for subspace approximation and k-median in nearly linear time. *CoRR*, abs/1912.12003, 2019.

[25] Z. Friggstad, M. Rezapour, and M. R. Salavatipour. Local search yields a PTAS for k-means in doubling metrics. *SIAM J. Comput.*, 48(2):452–480, 2019.

[26] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In

FOCS, pages 534–543. IEEE Computer Society, 2003.

[27] V. Guruswami and P. Indyk. Embeddings and non-approximability of geometric problems. In *SODA*, volume 3, pages 537–538, 2003.

[28] S. Har-Peled. On complexity, sampling, and ϵ-nets and ϵ-samples. In *Geometric approximation algorithms*, volume 173. American Mathematical Soc., 2011.

[29] S. Har-Peled and A. Kushal. Smaller coresets for k-median and k-means clustering. *Discret. Comput. Geom.*, 37(1):3–19, 2007.

[30] S. Har-Peled and S. Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, STOC '04, page 291–300. ACM, 2004.

[31] L. Huang, S. H. Jiang, J. Li, and X. Wu. Epsilon-coresets for clustering (with outliers) in doubling metrics. In *FOCS*, pages 814–825. IEEE Computer Society, 2018.

[32] L. Huang, S. H. Jiang, and N. K. Vishnoi. Coresets for clustering with fairness constraints. In *NeurIPS*, pages 7587–7598, 2019.

[33] L. Huang and N. K. Vishnoi. Coresets for clustering in Euclidean spaces: Importance sampling is nearly optimal. In *STOC*, pages 1416–1429. ACM, 2020.

[34] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 731–740, 2002.

[35] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang. On the placement of internet instrumentation. In *INFOCOM*, pages 295–304. IEEE Computer Society, 2000.

[36] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, pages 189–206. Amer. Math. Soc., 1984.

[37] Z. S. Karnin and E. Liberty. Discrepancy, coresets, and sketches in machine learning. In *COLT*, volume 99 of *Proceedings of Machine Learning Research*, pages 1975–1993. PMLR, 2019.

[38] M. Langberg and L. J. Schulman. Universal epsilon-approximators for integrals. In *SODA*, pages 598–607. SIAM, 2010.

[39] B. Li, M. J. Golin, G. F. Italiano, X. Deng, and K. Sohraby. On the optimal placement of web proxies in the internet. In *INFOCOM*, pages 1282–1290. IEEE Computer Society, 1999.

[40] M. Li, G. L. Miller, and R. Peng. Iterative row sampling. In *FOCS*, pages 127–136. IEEE Computer Society, 2013.

[41] Y. Liang, M.-F. Balcan, and V. Kanchanapally. Distributed PCA and k-means clustering. In *The Big Learning Workshop at NIPS*, volume 2013. Citeseer, 2013.

[42] M. Lucic, M. Faulkner, A. Krause, and D. Feldman. Training gaussian mixture models at scale via coresets. *The Journal of Machine Learning Research*, 18(1):5885–5909, 2017.

[43] A. Maalouf, I. Jubran, and D. Feldman. Fast and accurate least-mean-squares solvers. In *Advances in Neural Information Processing Systems*, pages 8305–8316, 2019.

[44] K. Makarychev, Y. Makarychev, and I. Razenshteyn. Performance of Johnson-Lindenstrauss transform for k-means and k-medians clustering. In *Proceedings of the 51st Annual Symposium on Theory of Computing*, pages 1027–1038, 2019.

[45] A. Munteanu, C. Schwiegelshohn, C. Sohler, and D. P. Woodruff. On coresets for logistic regression. In *NeurIPS*, pages 6562–6571, 2018.

[46] S. Narayanan and J. Nelson. Optimal terminal dimensionality reduction in Euclidean space. In *STOC*, pages 1064–1069. ACM, 2019.

[47] J. M. Phillips and W. M. Tai. Near-optimal coresets of kernel density estimates. *Discrete & Computational Geometry*, pages 1–21, 2019.

[48] M. J. Rattigan, M. E. Maier, and D. D. Jensen. Graph clustering with network structure indices. In *ICML*, volume 227 of *ACM International Conference Proceeding Series*, pages 783–790. ACM, 2007.

[49] M. Schmidt, C. Schwiegelshohn, and C. Sohler. Fair coresets and streaming algorithms for fair k-means. In *WAOA*, volume 11926 of *Lecture Notes in Computer Science*, pages 232–251. Springer, 2019.

[50] S. Shekhar and D. Liu. CCAM: A connectivity-clustered access method for networks and network computations. *IEEE Trans. Knowl. Data Eng.*, 9(1):102–119, 1997.

[51] C. Sohler and D. P. Woodruff. Strong coresets for k-median and subspace approximation: Goodbye dimension. In *FOCS*, pages 802–813. IEEE Computer Society, 2018.

[52] B. C. Tansel, R. L. Francis, and T. J. Lowe. State of the art—location on networks: a survey, part i and ii. *Management Science*, 29(4):482–497, 1983.

[53] M. Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *J. ACM*, 51(6):993–1024, 2004.

[54] M. Thorup. Quick k-Median, k-Center, and facility location for sparse graphs. *SIAM J. Comput.*, 34(2):405–432, 2005.

[55] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

[56] K. R. Varadarajan and X. Xiao. On the sensitivity of shape fitting problems. In *FSTTCS*, volume 18 of *LIPIcs*, pages 486–497. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

[57] M. L. Yiu and N. Mamoulis. Clustering objects on a spatial network. In *SIGMOD Conference*, pages 443–454. ACM, 2004.