Coresets for Clustering Using Noisy Comparisons and Few Distance Queries

Amir Carmel[*]
Weizmann Institute of Science

Robert Krauthgamer[†]
Weizmann Institute of Science

Abstract

Coresets are a fundamental tool for scaling clustering algorithms, but standard constructions assume full access to the pairwise distances. In many modern settings, access to distances is costly and thus limited, and algorithms must instead rely on inexpensive comparison queries. We study coreset construction in the Rank-Measure (RM) model, where the algorithm can make noisy comparison queries to a quadruplet oracle alongside a small number of distance queries. In this model, we present the first algorithm to construct an ϵ-coreset for (k, z)-clustering in Euclidean space. It matches the coreset-size bounds known in the classical model, while using only polylogarithmically many distance queries in the input size. Our approach is based on a uniform-sampling framework [Chen, SICOMP 2009], which decomposes the input into geometric rings and samples uniformly within each ring. We implement this approach using primarily noisy comparison queries, and then compress the result using a standard full-access coreset construction. In contrast, prior work in oracle-based clustering achieved only constant-factor guarantees. We complement our theoretical findings with experiments that demonstrate strong empirical performance against full-access baselines while using significantly fewer distance computations.

1 Introduction

Clustering is a cornerstone of unsupervised learning and a fundamental tool in data analysis, used to uncover structure in high-dimensional or large-scale datasets and as a preprocessing step for downstream tasks like retrieval, summarization, and learning. A central formulation is the (k, z)-*clustering* problem in Euclidean space (ℓ_2 norm): given a set of points $P \subseteq \mathbb{R}^d$, $|P| = n$, and parameters $k, z \geq 1$, find a set $C \subseteq \mathbb{R}^d$ of k points, called *centers*, that minimizes

$$\mathrm{cost}_z(C, P) := \sum_{p \in P} \mathrm{dist}(p, C)^z,$$

where $\mathrm{dist}(p, C) := \min_{c \in C} \mathrm{dist}(p, c)$. The cases $z = 1$ and $z = 2$ are the classical k-median and k-means objectives.

In the classical formulation, the input points are given as vectors in $\mathbb{R}^d$ and pairwise distances can be computed directly. Recent advances in machine learning have driven large-scale similarity models, particularly for non-metric data such as text, images, and video, where the similarity measure is not given a priori but is produced by an embedding model. As these models grow in complexity, the computational cost of pairwise distance evaluations increase rapidly. A common practical workaround is to pair an accurate expensive embedding with a cheaper but cruder one [Silwal et al.,

[*]Email: amir6423@gmail.com

[†]The Harry Weinrebe Professorial Chair of Computer Science. Work partially supported by the Israel Science Foundation grant #1336/23. Email: robert.krauthgamer@weizmann.ac.il

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

2023, Xu et al., 2024, Bateni et al., 2024]. The two embeddings induce different geometries, so their distances are not directly comparable, but they typically agree on the *relative* ordering of pairs. A separate but related scenario arises when similarity is supplied by human annotators, who can usually answer comparison questions of the form "is A closer to B, or is C closer to D?" but cannot return numerical distances [Tamuz et al., 2011, Law et al., 2013, Addanki et al., 2021].

Both scenarios are well captured by an *oracle-based* access model, where the algorithm queries an oracle instead of computing distances. The ***R-model***, introduced by Addanki et al. [2021], gives the algorithm access only to a noisy quadruplet oracle that, given two pairs of points, reports which pair is closer, subject to some failure probability $\varphi < \frac{1}{2}$. The noise is persistent, so repeated queries on the same pair return the same answer and a wrong answer cannot be corrected by re-querying. The *Rank-Measure (RM) model* extends the R-model with a distance oracle that returns $\mathrm{dist}(u, v)$ exactly. The goal in the RM-model is to lean on cheap quadruplet queries and make as few distance queries as possible (considered the primary complexity measure).

These models have been studied for k-center, k-median, and k-means clustering [Addanki et al., 2021, Galhotra et al., 2024, Raychaudhury et al., 2025, 2026]. In the RM-model, $O(1)$-approximation algorithms are known with sublinear complexity of distance queries. The R-model alone is severely limited. Addanki et al. [2021] obtained an $O(1)$-approximation for k-center, but only when the optimal clusters have size $\Omega(\sqrt{n})$; for k-median and k-means, no $o(n)$-approximation is possible in this model even with a noise-free quadruplet oracle, and any such approximation must return at least $2k - 1$ centers [Galhotra et al., 2024, Raychaudhury et al., 2026].

A central tool for handling clustering instances at scale is a *coreset*, a small weighted subset of the input that summarizes the original dataset for the purposes of clustering. Concretely, an ϵ-coreset for the (k, z)-clustering problem is a weighted set $S \subseteq P$ with weights $w : S \to \mathbb{R}_{>0}$ such that for every candidate center set $C \subseteq \mathbb{R}^d$ with $|C| \leq k$,

$$\mathrm{cost}_z(C, S) := \sum_{s \in S} w(s)\ \mathrm{dist}(s, C)^z \in (1 \pm \epsilon)\mathrm{cost}_z(C, P). \tag{1}$$

This is sometimes called a *strong coreset*, since the cost is preserved for every center set. Coresets are well understood for (k, z)-clustering in the classical model, where current constructions in $\mathbb{R}^d$ achieve size $2^{O(z \log z)}\epsilon^{-2}k \cdot \min(\epsilon^{-z}, k) \cdot \mathrm{polylog}(\epsilon^{-1}k)$,[3] which is independent of both n and d, and moreover tight up to the polylogarithmic term [Cohen-Addad et al., 2022, Huang et al., 2024]. This allows the desired clustering algorithm to run in time independent of the input size.

Coresets are a particularly useful technique in the RM-model, since they reduce the original instance to a small subset on which the problem can then be solved with few distance queries. Algorithms that construct an $O(1)$-coreset (defined as in (1) with $1 \pm \epsilon$ replaced by a constant factor) have been utilized as an intermediate step to solve the clustering objective [Galhotra et al., 2024, Raychaudhury et al., 2025, 2026]. Whether an ϵ-coreset (rather than an $O(1)$-coreset) is achievable in this model has remained open.

1.1 Our contribution

Our main result is an algorithm in the RM-model, that constructs an ϵ-coreset for (k, z)-clustering, whose size matches the best bounds known in the classical model, using a number of distance queries that is only polylogarithmic in n and polynomial in k/ϵ. We expect our techniques to extend to other access models where exact distances are expensive.

Theorem 1.1 (Main result)**.** *Given an n-points dataset $P \subset \mathbb{R}^d$ accessible via the RM-model with error rate $\varphi < 1/4$, and parameters $k, z, \epsilon, \delta > 0$,*[4] *one can construct, with probability at least $1 - \delta$, an ϵ-coreset S for (k, z)-clustering of size*

$$|S| \leq 2^{O(z \log z)}\epsilon^{-2}k \cdot \min(\epsilon^{-z}, k) \cdot \mathrm{polylog}(\epsilon^{-1}k) \cdot \log(1/\delta),$$

using $O(kn \operatorname{polylog} n)$ quadruplet queries and the number of distance queries is at most

$$2^{O(z \log z)}\epsilon^{-2}k^2 \cdot \min(\epsilon^{-2}, d) \cdot \log^2 n \cdot \mathrm{polylog}(\epsilon^{-1}k \log n).$$

[3] Another known bound $2^{O(z)}\epsilon^{-2}k^{\frac{2z+2}{z+2}}\ \mathrm{polylog}(\epsilon^{-1}k)$ is better in some parameter regime [Huang et al., 2024].

[4] While this is not a barrier to our technique, we assume $1/\delta \leq \mathrm{poly}(n)$.

Prior to this work, coreset results in the RM-model only guaranteed $O(1)$-approximation. Raychaudhury et al. [2026] adapt the framework of Har-Peled and Mazumdar [2004] to the R-model (improving on earlier construction of Galhotra et al. [2024] in the RM-model), obtaining for general (k, z)-clustering a set $C^0 \subseteq P$ of $O(k \operatorname{polylog} n)$ representatives together with a mapping $\mathcal{M} : P \to C^0$ whose induced clustering cost is within a constant factor of the optimum. Such a mapping has been used as a primitive to build coresets in classical settings [Har-Peled and Mazumdar, 2004, Chen, 2009, Bateni et al., 2014] and is sometimes called a *mapping coreset*. For k-median on metrics of bounded doubling dimension, they further obtain an ϵ-coreset, but its size depends on n, and the dependence on ϵ is exponential in the doubling dimension, which for our $\mathbb{R}^d$ is $O(d)$.

We complement our theoretical results with three experiments on real-world data using two embedding models (expensive and cheap) to instantiate the quadruplet and distance oracles. Our main experiment compares our construction against classical coreset baselines that have full direct access to the distances; we find that our coreset achieves competitive approximation guarantees despite making only a small number of distance queries and relying mostly on cheap quadruplet queries. We additionally vary the quadruplet noise rate φ, and find that the construction is effective for a wide range of values, even beyond the threshold $\varphi < 1/4$ from the theoretical analysis. We also verify empirically that the number of distance queries grows with n rather slowly.

1.2 Technical overview

Our algorithm has two phases. Phase 1 constructs an intermediate $(\epsilon/3)$-coreset S_1 of sublinear size while mostly relying on quadruplet queries. Phase 2 applies to S_1 an off-the-shelf full-access construction of $(\epsilon/3)$-coreset, which uses only $|S_1|$ distance queries and compresses S_1 into a final coreset S_2 of optimal size. Since coresets are composable, S_2 is an ϵ-coreset for the original input P. The main technical contribution clearly lies in Phase 1.

Our approach for constructing the $(\epsilon/3)$-coreset S_1 in the RM-model is based on the uniform-sampling framework of Chen [2009], whose analysis was later refined by Braverman et al. [2022]. It partitions the input into geometric rings around the centers of an $O(1)$-approximate solution, and samples uniformly within each ring. (In each ring, the ratio between the outer and inner radius is 2.) The key advantage is that this approach requires only limited information about the data points. Indeed, to implement it in the RM-model we need to (i) produce $O(1)$-approximate centers and partition the points accordingly; (ii) compute each cluster's radius; and (iii) classify every point into its correct ring. Nevertheless, these tasks are non-trivial when distance queries are not freely available, and moreover quadruplet queries are noisy.

For (i), we combine the algorithm ALG-G of Raychaudhury et al. [2026], which produces an $O(1)$-mapping coreset in the R-model, with the Mettu–Plaxton $O(1)$-approximation algorithm [Mettu and Plaxton, 2004]. We first run ALG-G on P to obtain a set of $O(k \operatorname{polylog} n)$ representatives, together with a mapping from each point in P to its representative. We then run Mettu–Plaxton on the resulting representatives to extract exactly k centers C, and partition P into clusters by composing the ALG-G mapping with C. Combining the two requires bounding both the cost of C on P (Lemma 4.3) and the cost of the induced partition (Lemma 4.4), each of which feeds into the Phase 1 correctness argument.

For the cluster radii (ii) and the ring classification (iii), we lean on the approximate sorting result of Geissmann et al. [2025], which, under persistent comparison noise, produces an ordering of points by distance to their cluster center with maximum dislocation $D = O(\log n)$, using only quadruplet queries. In fact, task (iii) is the core algorithmic subroutine of the entire pipeline. Naively computing $\operatorname{dist}(p, c_i)$ for every p would take $O(n)$ distance queries. Our algorithm relies instead on the approximate ordering and the dislocation bound: it performs binary search for each ring boundary in the ordering, resolves a small ambiguity region around it with distance queries, and classifies the remaining points by position alone. The full procedure (Algorithm 1) uses only $O(L_i \log n)$ distance queries per cluster, where $L_i = O(\log n + z^2 + z \log(1/\epsilon))$ is the number of rings of the i-th cluster.

In addition, we need a per-ring sampling guarantee that is strong enough so that aggregation across the rings would yield a coreset for P. This is established in Theorem 4.8, whose proof relies on the result of Braverman et al. [2022], combined with the terminal embedding of Narayanan and Nelson [2019] and the iterative size-reduction technique [Braverman et al., 2021, Cohen-Addad

et al., 2021a] to control the dependence on the ambient dimension. Aggregating the samples from all the rings then yields the intermediate coreset S_1 (Lemma 4.9), and completes Phase 1.

Finally, Phase 2 compresses S_1 into S_2 of optimal size, which completes the construction. We provide a step-by-step description of the entire algorithm in Section 3, and its formal analysis appears in Section 4. Due to space constraints all proofs are deferred to the appendix.

1.3 Related work

Triplet and quadruplet distance-comparison oracles have been used as the basic access model across a range of problems other than (k, z)-clustering, including similarity-kernel and fairness-metric learning [Tamuz et al., 2011, Ilvento, 2020], intrinsic dimension estimation [Kleindessner and von Luxburg, 2015], kernel construction [Kleindessner and von Luxburg, 2017a], medoid estimation and outlier identification [Kleindessner and von Luxburg, 2017b], nearest-neighbor search [Haghiri et al., 2017], classification and regression [Haghiri et al., 2018, Hopkins et al., 2020], hierarchical clustering [Ghoshdastidar et al., 2019], correlation clustering [Ukkonen, 2017], and clustering of stable instances [Bianchi and Penna, 2021].

A fundamental primitive underlying many comparison-based algorithms is sorting under persistent comparison errors. The model was introduced by Braverman and Mossel [2008], who observed that exact sorting is impossible in this regime and posed the natural objective of minimizing the dislocation of elements relative to their true rank. A sequence of works has progressively improved both the running time and the dislocation guarantees [Klein et al., 2011, Geissmann et al., 2017, 2020, 2019], culminating in the $O(n \log n)$-time algorithm of Geissmann et al. [2025], which achieves $O(\log n)$ maximum dislocation and further shows that no algorithm can achieve $o(\log n)$ maximum dislocation with high probability.

Another line of work studies center-based clustering under oracle-access models that differ from ours. Ashtiani et al. [2016] introduce the same-cluster query (SCQ) model, in which the oracle answer queries of the form "do points u, v belong to the same optimal cluster?", obtaining a polynomial-time algorithm for k-means under a margin assumption; Ailon et al. [2018] extend this to a $(1 + \epsilon)$-approximation algorithm without margin, and Huleihel et al. [2019] study the model for overlapping clusters. More closely related to our setting, Bateni et al. [2024] introduce the *weak-strong distance oracle* model, in which a cheap noisy distance oracle is paired with an expensive exact one (called weak and strong oracles), and the goal is to minimize the number of strong-oracle queries; they obtain $O(1)$-approximations for k-center, k-median, and k-means. Subsequent work has improved the query complexity [Pradhan et al., 2026] and extended the model to fair k-clustering [Braverman et al., 2025].

Coresets for (k, z)-clustering have been the subject of a long line of work, including [Har-Peled and Mazumdar, 2004, Feldman and Langberg, 2011, Sohler and Woodruff, 2018, Huang and Vishnoi, 2020, Cohen-Addad et al., 2021b, 2022, Huang et al., 2024], with modern constructions matching the known lower bounds up to polylogarithmic factors. Most relevant to our setting is the line of *uniform-sampling* coreset constructions, which originated in the work of Chen [2009], that proposed decomposing the metric space into "rings" and sampling uniformly from each ring. This approach was subsequently refined by Braverman et al. [2022], Cohen-Addad et al. [2021a], but the overall sampling distribution remains non-uniform across rings. Coreset constructions that truly employ uniform sampling require restrictions on the input and weaker coreset guarantees [Huang et al., 2023, Carmel and Krauthgamer, 2026].

2 Preliminaries

Throughout, $P \subset \mathbb{R}^d$ is the input point set, and its optimal cost is denoted by $\mathrm{OPT} := \min_{|C| \le k} \mathrm{cost}_z(C, P)$. For sets $U, W \subseteq P$, we write $E(U, W) := \{\{u, w\} : u \in U, w \in W, u \neq w\}$ for the set of unordered pairs with one endpoint in each. The quadruplet oracle's error probability $\varphi < 1/4$ is a fixed constant; constants hidden in $O(\cdot)$ notation may depend on φ. In the *R-model*, access to distances is entirely replaced by a noisy *quadruplet oracle*. For an error parameter $\varphi \in [0, 1/4)$, the oracle $\widetilde{Q} : E \times E \to \{\text{Yes}, \text{No}\}$ takes two pairs of points (edges) $e_1 = \{v_1, v_2\}$, $e_2 = \{v_3, v_4\}$ and answers whether $\mathrm{dist}(v_1, v_2) \le \mathrm{dist}(v_3, v_4)$, correctly with probability at least $1 - \varphi$. The noise is independent across distinct queries and persistent: repeated calls on the same

pair always return the same answer. The *RM-model* augments the R-model with a *distance oracle*: given $u, v \in P$, it returns $\mathrm{dist}(u, v)$ exactly.

We will repeatedly use the following relaxed triangle inequality for z-th power of distances. Equation (2) is from [Makarychev et al., 2019] and Equation (3) from [Cohen-Addad et al., 2021b].

Lemma 2.1 (Triangle inequality for powers)**.** *Let a, b, c be points in a metric space $(\mathcal{X}, \mathrm{dist})$, let $S \subseteq \mathcal{X}$ and let z be a positive integer. Then for all $t > 0$,*

$$\mathrm{dist}(a,b)^z \le (1+t)^{z-1}\,\mathrm{dist}(a,c)^z + \left(\frac{1+t}{t}\right)^{z-1}\mathrm{dist}(b,c)^z, \tag{2}$$

$$|\,\mathrm{dist}(a,S)^z - \mathrm{dist}(b,S)^z| \le t\cdot\mathrm{dist}(a,S)^z + \left(\frac{2z+t}{t}\right)^{z-1}\mathrm{dist}(a,b)^z. \tag{3}$$

Coresets also compose: if S_1 is a coreset for P and S_2 is a coreset for S_1, then S_2 is a coreset for P, with the errors essentially added.

Lemma 2.2 (Coreset composition)**.** *If S_1 is an ϵ_1-coreset for P and S_2 is an ϵ_2-coreset for S_1, then S_2 is an $(\epsilon_1 + \epsilon_2 + \epsilon_1\epsilon_2)$-coreset for P.*

3 Algorithm description

The algorithm has two phases. Phase 1, in which our main contribution lies, constructs an intermediate $(\epsilon/3)$-coreset S_1 of size $2^{O(z\log z)}\epsilon^{-2}k^2\min(d,\epsilon^{-2})\log^2 n\cdot\mathrm{polylog}(k\epsilon^{-1}\log n)$, using $O(kn\ \mathrm{polylog}\ n)$ quadruplet queries and a sublinear number of distance queries. Phase 2 further compresses S_1 by feeding it into an off-the-shelf full-access coreset construction, producing the final $(\epsilon/3)$-coreset S_2 of optimal size and using only $|S_1|$ additional distance queries. Coreset composition (Lemma 2.2) then yields an ϵ-coreset for the original input P. Phase 1 proceeds through the following steps.

Step 1: Constant factor approximation centers. Similarly to standard coreset construction algorithms, we begin by obtaining an $O(1)$-approximate set of k centers together with an assignment of points to centers. To achieve this in the RM-model we combine two components. The algorithm ALG-G of Raychaudhury et al. [2026] outputs, in the R-model, a set C^0 of $O(k\ \mathrm{polylog}\ n)$ representatives and a mapping $\mathcal{M} : P \to C^0$ whose total cost is within a constant factor of OPT, using $O(kn\ \mathrm{polylog}\ n)$ quadruplet queries. To reduce C^0 to k centers, we run the Mettu–Plaxton constant-factor algorithm [Mettu and Plaxton, 2004] on the set C^0 to extract exactly k centers $C = \{c_1, \ldots, c_k\}$. This requires $O(k^2\ \mathrm{polylog}\ n)$ distance queries. We then define a partitioning $P_1, \ldots, P_k$ of P, where each point $p \in P$ is assigned to the center $c_i \in C$ closest to $\mathcal{M}(p)$.

Step 2: Approximate sorting using the quadruplet oracle. In order to execute subsequent steps without relying on distance queries, we exploit the PROBSORT algorithm of Geissmann et al. [2025]. Under persistent noise with error rate $\varphi < 1/4$, PROBSORT produces, with high probability, an ordering with *maximum dislocation* $D = O(\log n)$, meaning every element lies within D positions of its true rank. We apply PROBSORT to the kn point-center pairs in $E(C, P)$, obtaining a global ordering π. For each cluster P_i, let π_i denote the induced ordering on the pairs in $E(c_i, P_i)$, which inherits dislocation at most D. This step uses $O(kn\log n)$ quadruplet queries and 0 distance queries.

Step 3: Identify cluster radii. For each cluster P_i we identify the maximum distance $d^{(i)}_{\max} := \max_{p\in P_i}\mathrm{dist}(p, c_i)$. Since π_i has dislocation D, the true farthest point appears among the last D positions of π_i. Using $O(k\log n)$ distance queries we identify $d^{(i)}_{\max}$ for all $1 \le i \le k$.

Step 4: Ring decomposition and classification. We partition each cluster P_i into a *close set* P^{cl}_i, whose points will be snapped to the center c_i, and $L_i = O(\log n + z^2 + z\log(1/\epsilon))$ rings $R_{i,1}, \ldots, R_{i,L_i}$. Each ring is centered at c_i and has ratio 2, meaning that the maximum and minimum distances from c_i are within factor 2. The boundaries are distance thresholds derived from $d^{(i)}_{\max}$. To classify points without querying $\mathrm{dist}(p, c_i)$ for every p, we exploit π_i. Each threshold corresponds to a position in π_i that we can locate within D via binary-search-with-noise, a folklore technique using $O(\log n + D)$ queries. The $O(D)$-wide ambiguity zone around each boundary is resolved

with exact distance queries, and all remaining points are classified by position alone. Overall, every point is classified correctly using $O(L_i \log n)$ distance queries per cluster (Lemma 4.7).

Step 5: Per-ring sampling. Building on uniform-sampling techniques from the coreset literature, we show that for a ratio-2 ring R around q, a uniform sample $\Omega \subseteq R$ of $s = 2^{O(z \log z)} \cdot \widetilde{O}(\epsilon^{-2}) \cdot k \cdot \min(d, \epsilon^{-2})$ points,[5] where each sampled point is reweighted to $|R|/s$, approximates the ring's cost for every $A \subset \mathbb{R}^d$ with $|A| \le k$, up to additive error $\epsilon(\mathrm{cost}_z(q, R) + \mathrm{cost}_z(A, R))$. We apply this independently to each ring $R_{i,\ell}$. The close set P_i^{cl} is snapped to c_i, which is reweighted to $|P_i^{\mathrm{cl}}|$. The intermediate coreset is then $S_1 := \bigcup_i (\{c_i\} \cup \bigcup_\ell \Omega_{i,\ell})$. This sampling step uses no distance queries. This concludes Phase 1, after which S_1 is an $(\epsilon/3)$-coreset for P.

Step 6: Phase 2 compression. To further compress S_1 and obtain a state-of-the-art coreset size, we feed it into the offline coreset construction of Cohen-Addad et al. [2022, 2021b], supplying the required distances $\mathrm{dist}(c_i, p)$ for $p \in S_1$ with $|S_1|$ distance queries. The output is the final coreset S_2, of size $2^{O(z \log z)} \epsilon^{-2} k \cdot \min(\epsilon^{-z}, k) \cdot \mathrm{polylog}(\epsilon^{-1} k)$, with high constant probability. Since S_1 is a coreset of P and S_2 is a coreset of S_1, Lemma 2.2 gives that S_2 is a coreset of P.

4 Analysis

4.1 Step 1: Constant factor approximation centers

The key tool for this step is the algorithm ALG-G of Raychaudhury et al. [2026], which produces a constant-factor mapping in the R-model.

Theorem 4.1 (ALG-G; Theorem 3.1 of Raychaudhury et al. [2026])**.** *There exists a randomized algorithm* ALG-G *that, given an n-point metric space (P, dist) accessible via the R-model and parameters $k, z \in \mathbb{Z}_+$, computes with high probability a set $C^0 \subseteq P$ with $|C^0| = O(k \,\mathrm{polylog}\, n)$ and a mapping $\mathcal{M} : P \to C^0$ satisfying $\sum_{p \in P} \mathrm{dist}(p, \mathcal{M}(p))^z \le \alpha_0 \cdot \mathrm{OPT}$ for a constant $\alpha_0 > 0$. The algorithm uses $O(kn \,\mathrm{polylog}\, n)$ queries to the quadruplet oracle (and zero distance queries).*

After using ALG-G to compute C^0, we run on it the Mettu–Plaxton algorithm to extract exactly k centers. The original statement of Mettu and Plaxton [2004] is only for k-median ($z = 1$); the extension to general z is via the relaxed triangle inequality (2), cf. [Huang and Vishnoi, 2020, Cohen-Addad et al., 2021b].

Theorem 4.2 (Mettu and Plaxton [2004])**.** *Given a weighted finite metric space (Q, dist, w) with $|Q| = m$ and an integer $k \ge 1$, the Mettu–Plaxton algorithm computes, with high probability, a k-center set $C \subseteq Q$ satisfying $\mathrm{cost}_z(C, Q_w) \le \beta_z \cdot \min_{|A| \le k} \mathrm{cost}_z(A, Q_w)$, where $\beta_z \ge 1$ is a constant depending only on z. The algorithm uses $O(mk)$ distance queries.*

We now describe Step 1 in detail. Run ALG-G$(P, k, z) \to (C^0, \mathcal{M})$, obtaining a set $C^0 \subseteq P$ where $|C^0| = O(k \,\mathrm{polylog}\, n)$ and a mapping $\mathcal{M} : P \to C^0$ such that $\sum_p \mathrm{dist}(p, \mathcal{M}(p))^z \le \alpha_0 \mathrm{OPT}$. Construct weights $w_{\mathcal{M}}(c) = |\{p \in P : \mathcal{M}(p) = c\}|$ for each $c \in C^0$. We write $\mathrm{cost}_z(A, C^0_w)$ for the weighted cost $\sum_{c \in C^0} w_{\mathcal{M}}(c) \cdot \mathrm{dist}(c, A)^z$.

Run Mettu–Plaxton (Lemma 4.2) on the weighted set $(C^0, w_{\mathcal{M}})$ using the distance oracle to obtain $C = \{c_1, \ldots, c_k\}$. Define the partition $P_i := \{p \in P : c_i = \arg\min_{c \in C} \mathrm{dist}(\mathcal{M}(p), c)\}$ (breaking ties by index), the cluster size $n_i := |P_i|$, and the average cost $V_i := \mathrm{cost}_z(c_i, P_i)/n_i$.

Lemma 4.3. $\mathrm{cost}_z(C, P) \le \alpha \mathrm{OPT}$, *for* $\alpha = 2^{z-1}\alpha_0 + 4^{z-1}\beta_z(1 + \alpha_0)$.

Lemma 4.4. $\sum_i \mathrm{cost}_z(c_i, P_i) \le \alpha' \mathrm{OPT}$, *for* $\alpha' = (2^{z-1} + 4^{z-1})\alpha_0 + 4^{z-1}\alpha$.

4.2 Step 2: Approximate sorting using the quadruplet oracle

We apply the sorting algorithm of Geissmann et al. [2025].

Lemma 4.5 (PROBSORT; Geissmann et al., 2025; see also Lemma 2.1 of Raychaudhury et al., 2026)**.** *Let (P, dist) be a metric with $|P| = n$ and $E \subseteq E(P)$ be its edge set. Suppose $\widetilde{Q}$*

[5] Throughout, $\widetilde{O}(x)$ hides polylogarithmic factors in x.

is a quadruplet oracle with noise $\varphi < 1/4$. There exists an algorithm ProbSort *that, with probability $1 - n^{-4/3}$, outputs an ordering π_E with maximum dislocation $D = O(\log n)$ using $O(\max(|E|, n) \log n)$ queries to $\widetilde{Q}$.*

Run ProbSort on the edge set $E(C, P)$, which contains kn edges, to obtain with high probability a global ordering π with maximum dislocation $D = O(\log n)$. For each cluster P_i, let π_i denote the induced ordering on the pairs in $E(c_i, P_i)$, which inherits dislocation at most D.

4.3 Step 3: Identify cluster radii

For each cluster P_i with center c_i, determine the maximum distance $d_{\max}^{(i)} := \max_{p \in P_i} \operatorname{dist}(p, c_i)$. Since π_i has dislocation D, the farthest point in P_i appears among the last D positions of π_i. We query $\operatorname{dist}(c_i, \pi_i[j])$ for all $j \in \{n_i - D, \ldots, n_i\}$ and set $d_{\max}^{(i)} := \max_j \operatorname{dist}(c_i, \pi_i[j])$.

Lemma 4.6. *$d_{\max}^{(i)} = \max_{p \in P_i} \operatorname{dist}(p, c_i)$. Moreover, $(d_{\max}^{(i)})^z \le \operatorname{cost}_z(c_i, P_i)$.*

4.4 Step 4: Ring decomposition and classification

We partition each cluster P_i into a *close set* P_i^{cl}, whose points will be snapped to the center c_i in Step 5, and L_i ratio-2 *rings* $R_{i,1}, \ldots, R_{i,L_i}$, from which we will sample uniformly in Step 5. The partition is defined by distance thresholds. Computing $\operatorname{dist}(p, c_i)$ for every p would require n_i distance queries; instead, we use the approximate sorted order π_i from Step 2 together with $O(L_i \log n)$ targeted distance queries to classify every point correctly (Algorithm 1).

Define $\hat{V}_i := (d_{\max}^{(i)})^z / n_i$; note that $\hat{V}_i \le V_i$ by Lemma 4.6. Set the close threshold

$$\tau_{\mathrm{cl}}^{(i)} := \frac{\hat{V}_i}{\alpha'} \Big(\frac{\epsilon_1}{3z} \Big)^z = \frac{1}{n_i \alpha'} \Big(\frac{d_{\max}^{(i)} \cdot \epsilon_1}{3z} \Big)^z, \qquad \epsilon_1 := \frac{\epsilon}{6 \cdot 2^z \cdot (1+\alpha')}. \tag{4}$$

The close set and rings are

$$P_i^{\mathrm{cl}} := \{p \in P_i : \operatorname{dist}(p, c_i)^z \le \tau_{\mathrm{cl}}^{(i)}\},$$
$$R_{i,\ell} := \{p \in P_i : 2^{\ell-1} \tau_{\mathrm{cl}}^{(i)} < \operatorname{dist}(p, c_i)^z \le 2^{\ell} \tau_{\mathrm{cl}}^{(i)}\}, \qquad \ell = 1, \ldots, L_i.$$

The number of rings is

$$\begin{aligned} L_i &= \big\lceil \log_2 \big((d_{\max}^{(i)})^z \,/\, \tau_{\mathrm{cl}}^{(i)} \big) \big\rceil = \big\lceil \log_2 \big(n_i \alpha' (3z/\epsilon_1)^z \big) \big\rceil \\ &= \big\lceil \log_2 \big(n_i \alpha' (18z \cdot 2^z \cdot (1+\alpha')/\epsilon)^z \big) \big\rceil = O\big(\log n + z^2 + z \log(1/\epsilon) \big). \end{aligned}$$

Since every $p \in P_i$ satisfies $\operatorname{dist}(p, c_i)^z \le (d_{\max}^{(i)})^z = 2^{L_i} \tau_{\mathrm{cl}}^{(i)}$, every point is in P_i^{cl} or in some ring $R_{i,\ell}$. The thresholds $\theta_0 := \tau_{\mathrm{cl}}^{(i)}, \theta_1 := 2\tau_{\mathrm{cl}}^{(i)}, \ldots, \theta_{L_i} := 2^{L_i} \tau_{\mathrm{cl}}^{(i)}$ are computed directly from Equation (4). To classify each point p into its ring, we determine which interval $(\theta_{\ell-1}, \theta_\ell]$ contains $\operatorname{dist}(p, c_i)^z$, using Algorithm 1. Since π_i is approximately sorted with maximum displacement D, finding a position within D of the true boundary is a folklore problem requiring $O(\log n + D)$ queries (e.g., [Narasimhan, 2023]). Algorithm 1 uses this in Step 4.1 to locate each boundary approximately, then resolves the $O(D)$-width ambiguity zone in Step 4.2 by querying exact distances, and assigns the remaining points by position alone in Step 4.3.

Lemma 4.7 (Correctness of Algorithm 1)**.** *Assuming π is approximately sorted with maximum displacement D, Algorithm 1 classifies every point $p \in P_i$ into its correct ring $R_{i,\ell}$ or P_i^{cl}. The total number of distance queries for cluster i is $O(L_i \cdot (\log n + D)) = O(L_i \log n)$.*

4.5 Step 5: Phase 1 coreset

The sampling guarantee for ratio-2 rings is the following theorem, building on Braverman et al. [2022] together with an iterative size-reduction argument.

Theorem 4.8 (Per-ring coreset for (k, z)-clustering)**.** *Let $q \in \mathbb{R}^d$ and $R \subset \mathbb{R}^d$ be a ratio-2 ring around q. A uniform sample $\Omega \subseteq R$ of*

$$s = 2^{O(z \log z)} \cdot k \cdot \min(d, \epsilon^{-2}) \cdot \widetilde{O}(\epsilon^{-2}) \cdot \operatorname{polylog}(1/\delta)$$

Algorithm 1 ClassifyRings$(P_i, c_i, \pi_i, \{\theta_\ell\}_{\ell=0}^{L_i})$

Require: Cluster P_i with center c_i; π_i approximately sorted with maximum displacement D; thresholds $\theta_0 < \cdots < \theta_{L_i}$.
Ensure: Ring label $\mathrm{ring}(p) \in \{0, 1, \ldots, L_i\}$ for each $p \in P_i$ (label 0 means close).

Step 4.1: Locate approximate boundary positions.
for $\ell = 0, 1, \ldots, L_i$ **do**
  Search for threshold θ_ℓ in π_i, querying $\mathrm{dist}(c_i, \pi_i[j])^z$ as needed.
  Let j_ℓ be the returned position. ▷ within D of true boundary; $O(\log n + D)$ queries
end for

Step 4.2: Resolve ambiguity zones.
for $\ell = 0, 1, \ldots, L_i$ **do**
  $Z_\ell \leftarrow \{j_\ell - D, \ldots, j_\ell + D\} \cap [1, n_i]$
  for each $j \in Z_\ell$ not already queried in Step 4.1 **do**
    Query and record $\mathrm{dist}(c_i, \pi_i[j])$. ▷ $O(D)$ new queries per boundary
  end for
end for

Step 4.3: Assign ring labels.
for $j = 1, \ldots, n_i$ **do**
  $p \leftarrow \pi_i[j]$
  if $j \in Z_\ell$ for some ℓ **then** ▷ $\mathrm{dist}(p, c_i)$ known from Steps 4.1–4.2
    Assign $\mathrm{ring}(p)$ by comparing $\mathrm{dist}(p, c_i)^z$ to the thresholds.
  else ▷ p is far from all boundaries
    Let ℓ' be the unique ring such that $j_{\ell'-1} + D < j$ and $j < j_{\ell'} - D$.
    Assign $\mathrm{ring}(p) \leftarrow \ell'$.
  end if
end for

points, each weighted $|R|/s$, satisfies with probability at least $1 - \delta$, for all $A \subset \mathbb{R}^d$ with $|A| \le k$,

$$\left|\mathrm{cost}_z(A, R) - \tfrac{|R|}{s}\mathrm{cost}_z(A, \Omega)\right| \le \epsilon\big(\mathrm{cost}_z(q, R) + \mathrm{cost}_z(A, R)\big).$$

We now construct S_1. For each cluster P_i:

- *Close points.* Assign weight $|P_i^{\mathrm{cl}}|$ to center c_i, accounting for all close points snapped to it.
- *Rings.* For each ring $R_{i,\ell}$, draw a uniform sample $\Omega_{i,\ell} \subseteq R_{i,\ell}$ of size

  $$s = 2^{O(z \log z)} \cdot k \cdot \min(d, \epsilon^{-2}) \cdot \widetilde{O}(\epsilon^{-2}) \cdot \log n$$

  as in Theorem 4.8 with precision ϵ_1 and failure probability $\delta = n^{-c}/(\sum_{i=1}^{k} L_i)$ for a constant $c \ge 1$. Weight each sampled point by $|R_{i,\ell}|/s$.

Define the intermediate coreset $S_1 := \bigcup_i \big(\{c_i\} \cup \bigcup_\ell \Omega_{i,\ell}\big)$.

Lemma 4.9. *With high probability, S_1 is an $(\epsilon/3)$-coreset for (k, z)-clustering on P.*

Let $L = \max_{1 \le i \le k} L_i$. The size of S_1 is

$$|S_1| \le k + k \cdot L \cdot s = 2^{O(z \log z)} \cdot k^2 \cdot \min(d, \epsilon^{-2}) \cdot \epsilon^{-2} \cdot \log^2 n \cdot \mathrm{polylog}(k\epsilon^{-1} \log n). \tag{5}$$

4.6 Step 6: Phase 2 compressed coreset

The compression in this step uses the offline coreset framework of Cohen-Addad et al. [2021b], with the improved analysis of Cohen-Addad et al. [2022] and Huang et al. [2024]. The required input is a weighted set together with the distances from each point to its assigned center in an $O(1)$-approximate clustering.

Theorem 4.10 (Theorem 4 of Cohen-Addad et al., 2022; Theorem 1.2 and Corollary 3.3 of Cohen-Addad et al., 2021b)**.** *Given a weighted set $C \subset \mathbb{R}^d$ with distances $\mathrm{dist}(q, c_i)$ known for each point q and its assigned center c_i from an $O(1)$-approximate clustering, and a failure probability $\delta > 0$, one can construct, with probability at least $1 - \delta$, an ϵ-coreset for (k, z)-clustering of size*

$$2^{O(z \log z)} \epsilon^{-2} k \cdot \min(\epsilon^{-z}, k) \cdot \mathrm{polylog}(k\epsilon^{-1}) \cdot \log(1/\delta).$$

After querying $\mathrm{dist}(c_i, p)$ for each $p \in S_1$, we apply Theorem 4.10 with precision $\epsilon/3$ and failure probability $\delta/2$, using the centers C from Step 1 as the $O(1)$-approximate clustering, to obtain S_2.

Lemma 4.11. *With probability $1 - \delta/2$, S_2 is an $(\epsilon/3)$-coreset for S_1, of size*

$$|S_2| = 2^{O(z \log z)} \cdot k \cdot \min(\epsilon^{-z}, k) \cdot \epsilon^{-2} \cdot \log(1/\delta) \cdot \mathrm{polylog}(k/\epsilon). \tag{6}$$

4.7 Algorithm correctness

Proof of Theorem 1.1. By Lemma 4.9, S_1 is an $(\epsilon/3)$-coreset for P. By Lemma 4.11, S_2 is an $(\epsilon/3)$-coreset for S_1. By coreset composition (Lemma 2.2) and the fact that $\epsilon/3 + \epsilon/3 + \epsilon^2/9 < \epsilon$, S_2 is an ϵ-coreset for P of the stated size. Steps 1–5 succeed with high probability and Step 6 succeeds with probability $1 - \delta/2$, so by the union bound, the algorithm succeeds with probability $1 - \delta$.

See Table 1 for a per-step query counts. Quadruplet queries are dominated by Step 1, giving a total of $O(kn \,\mathrm{polylog}\, n)$. Distance queries are dominated by Step 6, whose total is given by (5). □

Table 1: Per-step query counts. Step 1 dominates the quadruplet queries; Step 6 dominates the distance queries.

Step	Description	Quadruplet queries	Distance queries
1	Constant factor approximation centers	$O(kn \,\mathrm{polylog}\, n)$	$O(k^2 \,\mathrm{polylog}\, n)$
2	Approximate sorting (ProbSort)	$O(kn \log n)$	0
3	Identify cluster radii	0	$O(k \log n)$
4	Ring decomposition and classification	0	$O(kL \log n)$
5	Phase 1 coreset	0	0
6	Phase 2 compressed coreset	0	$\lvert S_1 \rvert$
	Total	$O(kn \,\mathrm{polylog}\, n)$	$\lvert S_1 \rvert$

5 Experiments

We implement our algorithm for the k-means problem ($z = 2$). Phase 2 is implemented via sensitivity sampling, which is optimal up to polylogarithmic factors for k-means [Bansal et al., 2024]. For sake of practicality, Phase 1 is implemented with two adaptations: The Mettu–Plaxton subroutine in Step 1 is replaced by k-means++, and the ProbSort algorithm in Step 2 is replaced by Merge-Sort. Although Merge-Sort has no theoretical guarantees under noisy comparisons, we assume throughout the experiments a dislocation bound $D = 30$.

We then run three experiments on real-world datasets, which are listed with their sizes in Table 2. For the embedding models, we use text encoders from the BGE family [Xiao et al., 2024], and image encoders MobileNetV3-Small [Howard et al., 2019] and DINOv2-Large [Oquab et al., 2024]. Each experimental result is the average over 10 statistically independent repetitions, with mean curves and shaded standard-deviation bands. The headline metric in Experiments 2 and 3 is the *empirical worst-case cost ratio*, defined as

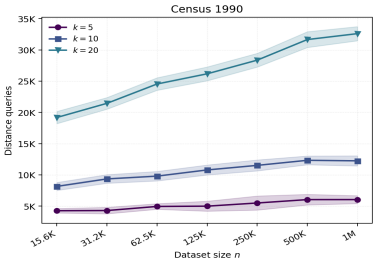
$$\max_{C \in \mathcal{C}} \frac{|\mathrm{cost}_z(C, P) - \mathrm{cost}_z(C, S)|}{\mathrm{cost}_z(C, P)},$$

where S is the coreset and the maximum is taken over a set $\mathcal{C}$ of candidate center sets generated by independent runs of k-means++ [Arthur and Vassilvitskii, 2007] on the full dataset. The idea is to empirically estimate the strong-coreset guarantee (1) by quantifying over a finite collection of candidate solutions, cf. [Schwiegelshohn and Sheikh-Omar, 2022]. All experiments were conducted on AMD EPYC 9354 nodes (32 physical cores per socket, 502 GB RAM) running AlmaLinux 8.10 with Python. The source code used to run the experiments is available at `https://github.com/amircarmel-lab/CoresetsRMModel`.

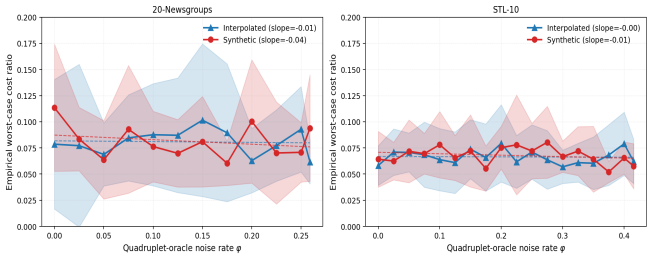
Experiment 1: Scaling of distance queries. We measure how the number of distance queries used by our coreset construction scales with n, on Census 1990, for $n \in \{15625, \ldots, 10^6\}$ and $k \in \{5, 10, 20\}$. Figure 1a shows that the count grows mildly with n, in line with the polylogarithmic dependence given by Theorem 1.1.

Table 2: Datasets used in the experiments, listed with their size n, true number of clusters k, and their cheap and expensive encoders, each annotated by its embedding dimension and the number of parameters in the model. The noise rate φ_{act} is the empirical disagreement rate between the cheap and expensive encoders.

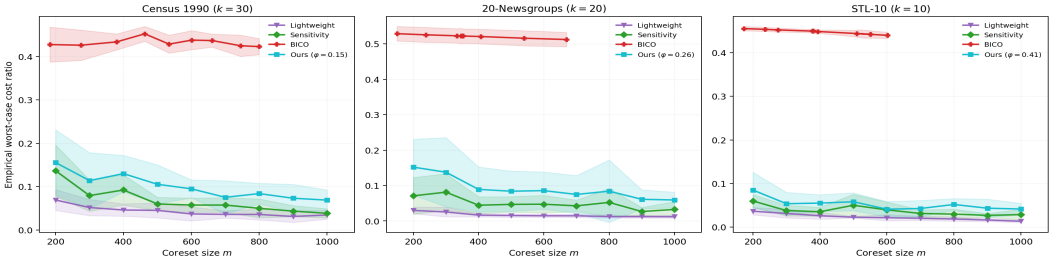
Dataset	n	k	**Cheap encoder**	**Expensive encoder**	φ_{act}
20-Newsgroups [Lang, 1995]	18,846	20	`bge-micro-v2` (384d, 17M)	`bge-large-v1.5` (1024d, 335M)	0.255
STL-10 [Coates et al., 2011]	13,000	10	MobileNetV3-S (576d, 2.5M)	DINOv2-L (1024d, 304M)	0.416
Census 1990 [Meek et al., 2001]	2.45M	N/A	—	—	—



(a) Number of distance queries vs. n (log-axis) on Census data for $k \in \{5, 10, 20\}$.



(b) Empirical worst-case cost ratio vs. φ on 20-Newsgroups (left) and STL-10 (right). Each panel shows synthetic and interpolated oracles.



(c) Empirical worst-case cost ratio vs. coreset size m on Census 1990 (left), 20-Newsgroups (middle), and STL-10 (right). Baseline algorithms have full access to pairwise distances; our algorithm uses a noisy quadruplet oracle ($\varphi = 0.15$ on Census, cheap encoder with φ_{act} noise on 20-Newsgroups and STL-10) plus a small number of distance queries.

Experiment 2: Robustness to quadruplet oracle noise rate. We measure how the empirical worst-case cost ratio of our coreset construction varies with the quadruplet oracle's noise rate φ, on two datasets, 20-Newsgroups ($k = 20$) and STL-10 ($k = 10$). We use two different oracle implementations: A *synthetic* oracle is built from the expensive embedding by flipping each pairwise comparison independently at rate φ (fixed across re-queries). An *interpolated* oracle routes each query to the cheap embedding with probability $\varphi/\varphi_{\text{act}}$ and to the expensive embedding with the remaining probability; this design controls the error between the two models. In both oracles, the expected effective noise rate is φ. Figure 1b shows that the cost ratio is nearly flat in φ across both oracles and both datasets, indicating that our coreset construction is robust to the noise rate, in fact even beyond the regime $\varphi < 1/4$ our analysis covers.

Experiment 3: Worst-case cost ratio against full-access baselines. We compare our coreset construction to three baseline algorithms that have full-access, BICO [Fichtenberger et al., 2013], Lightweight Coresets [Bachem et al., 2018], and Sensitivity Sampling [Feldman and Langberg, 2011], on three datasets, Census 1990 ($k = 30$, sample of size $n = 25{,}000$), 20-Newsgroups ($k = 20$), and STL-10 ($k = 10$). The baseline algorithms access the pairwise distances throughout, while our algorithm relies on a noisy quadruplet oracle plus a small number of distance queries. For Census we use a synthetic quadruplet oracle at $\varphi = 0.15$, and for 20-Newsgroups and STL-10 we use the quadruplet oracle induced by the cheap encoder. We vary the coreset size $m \in \{200, 300, \ldots, 1000\}$ and measure the empirical worst-case cost ratio. Figure 1c shows that our algorithm is competitive with the classical baselines, and exhibits only a modest gap in accuracy, despite having noisy and limited access to the distances. In fact, even a noiseless quadruplet oracle ($\varphi = 0$) is strictly weaker than full access, since it reports which of the two pairs is closer but never by how much.

References

Raghavendra Addanki, Sainyam Galhotra, and Barna Saha. How to design robust algorithms using noisy comparison oracle. *Proc. VLDB Endow.*, 14(10):1703–1716, 2021. doi:10.14778/3467861.3467862.

Nir Ailon, Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar. Approximate clustering with same-cluster queries. In *ITCS*, LIPIcs, pages 40:1–40:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICS.ITCS.2018.40.

David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *SODA*, pages 1027–1035. SIAM, 2007.

Hassan Ashtiani, Shrinu Kushagra, and Shai Ben-David. Clustering with same-cluster queries. In *NeurIPS*, pages 3216–3224, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/9597353e41e6957b5e7aa79214fcb256-Abstract.html.

Olivier Bachem, Mario Lucic, and Andreas Krause. Scalable k-means clustering via lightweight coresets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 1119–1127. Association for Computing Machinery, 2018. doi:10.1145/3219819.3219973.

Nikhil Bansal, Vincent Cohen-Addad, Milind Prabhu, David Saulpic, and Chris Schwiegelshohn. Sensitivity sampling for k-means: Worst case and stability optimal coreset bounds. In *FOCS*, pages 1707–1723. IEEE, 2024. doi:10.1109/FOCS61266.2024.00106.

MohammadHossein Bateni, Aditya Bhaskara, Silvio Lattanzi, and Vahab S. Mirrokni. Distributed balanced clustering via mapping coresets. In *NeurIPS*, pages 2591–2599, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/861dc9bd7f4e7dd3cccd534d0ae2a2e9-Abstract.html.

MohammadHossein Bateni, Prathamesh Dharangutte, Rajesh Jayaram, and Chen Wang. Metric clustering and MST with strong and weak distance oracles. In *COLT*, Proceedings of Machine Learning Research, pages 498–550. PMLR, 2024. URL https://proceedings.mlr.press/v247/bateni24a.html.

Enrico Bianchi and Paolo Penna. Optimal clustering in stable instances using combinations of exact and noisy ordinal queries. *Algorithms*, 14(2):55, 2021. doi:10.3390/A14020055.

Mark Braverman and Elchanan Mossel. Noisy sorting without resampling. In *SODA*, pages 268–276. SIAM, 2008. URL http://dl.acm.org/citation.cfm?id=1347082.1347112.

Vladimir Braverman, Shaofeng H.-C. Jiang, Robert Krauthgamer, and Xuan Wu. Coresets for clustering in excluded-minor graphs and beyond. In *SODA*, pages 2679–2696. SIAM, 2021. doi:10.1137/1.9781611976465.159.

Vladimir Braverman, Vincent Cohen-Addad, Shaofeng H.-C. Jiang, Robert Krauthgamer, Chris Schwiegelshohn, Mads Bech Toftrup, and Xuan Wu. The power of uniform sampling for coresets. In *FOCS*, pages 462–473. IEEE, 2022. doi:10.1109/FOCS54457.2022.00051.

Vladimir Braverman, Prathamesh Dharangutte, Shaofeng H.-C. Jiang, Hoai-An Nguyen, Chen Wang, Yubo Zhang, and Samson Zhou. Relative error fair clustering in the weak-strong oracle model. In *ICML*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/braverman25a.html.

Amir Carmel and Robert Krauthgamer. Stable coresets: Unleashing the power of uniform sampling. In *14th International Conference on Learning Representations, ICLR 2026*, 2026. URL https://openreview.net/forum?id=sOpAa8iR0A.

Ke Chen. On coresets for k-median and k-means clustering in metric and Euclidean spaces and their applications. *SIAM J. Comput.*, 39(3):923–947, 2009. doi:10.1137/070699007.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pages 215–223. PMLR, 2011. URL https://proceedings.mlr.press/v15/coates11a.html.

Vincent Cohen-Addad, David Saulpic, and Chris Schwiegelshohn. Improved coresets and sublinear algorithms for power means in Euclidean spaces. In *NeurIPS*, pages 21085–21098. Curran Associates, Inc., 2021a. URL https://proceedings.neurips.cc/paper/2021/hash/b035d6563a2adac9f822940c145263ce-Abstract.html.

Vincent Cohen-Addad, David Saulpic, and Chris Schwiegelshohn. A new coreset framework for clustering. In *STOC*, pages 169–182. ACM, 2021b. doi:10.1145/3406325.3451022.

Vincent Cohen-Addad, Kasper Green Larsen, David Saulpic, and Chris Schwiegelshohn. Towards optimal lower bounds for k-median and k-means coresets. In *STOC*, pages 1038–1051. ACM, 2022. doi:10.1145/3519935.3519946.

Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *STOC*, pages 569–578. ACM, 2011. doi:10.1145/1993636.1993712.

Hendrik Fichtenberger, Marc Gillé, Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. BICO: BIRCH meets coresets for k-means clustering. In *ESA*, Lecture Notes in Computer Science, pages 481–492. Springer, 2013. doi:10.1007/978-3-642-40450-4_41. URL https://doi.org/10.1007/978-3-642-40450-4_41.

Sainyam Galhotra, Rahul Raychaudhury, and Stavros Sintos. k-clustering with comparison and distance oracles. *Proc. ACM Manag. Data*, 2(5):212:1–212:26, 2024. doi:10.1145/3695830.

Barbara Geissmann, Stefano Leucci, Chih-Hung Liu, and Paolo Penna. Sorting with recurrent comparison errors. In *ISAAC*, LIPIcs, pages 38:1–38:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICS.ISAAC.2017.38.

Barbara Geissmann, Stefano Leucci, Chih-Hung Liu, and Paolo Penna. Optimal sorting with persistent comparison errors. In *ESA*, LIPIcs, pages 49:1–49:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ESA.2019.49.

Barbara Geissmann, Stefano Leucci, Chih-Hung Liu, and Paolo Penna. Optimal dislocation with persistent errors in subquadratic time. *Theory Comput. Syst.*, 64(3):508–521, 2020. doi:10.1007/S00224-019-09957-5.

Barbara Geissmann, Stefano Leucci, Chih-Hung Liu, and Paolo Penna. An optimal sorting algorithm for persistent random comparison faults, 2025. URL https://arxiv.org/abs/2508.19785.

Debarghya Ghoshdastidar, Michaël Perrot, and Ulrike von Luxburg. Foundations of comparison-based hierarchical clustering. In *NeurIPS*, pages 7454–7464, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/a76da37101dffabe00e5d636c01719b6-Abstract.html.

Siavash Haghiri, Debarghya Ghoshdastidar, and Ulrike von Luxburg. Comparison-based nearest neighbor search. In *AISTATS*, Proceedings of Machine Learning Research, pages 851–859. PMLR, 2017. URL http://proceedings.mlr.press/v54/haghiri17a.html.

Siavash Haghiri, Damien Garreau, and Ulrike von Luxburg. Comparison-based random forests. In *ICML*, Proceedings of Machine Learning Research, pages 1866–1875. PMLR, 2018. URL http://proceedings.mlr.press/v80/haghiri18a.html.

Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *STOC*, pages 291–300. ACM, 2004. doi:10.1145/1007352.1007400.

Max Hopkins, Daniel Kane, Shachar Lovett, and Gaurav Mahajan. Noise-tolerant, reliable active classification with comparison queries. In *Conference on Learning Theory*, pages 1957–2006. PMLR, 2020.

Andrew Howard, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, Yukun Zhu, Ruoming Pang, Hartwig Adam, and Quoc Le. Searching for MobileNetV3 . In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1314–1324. IEEE Computer Society, 2019. doi:10.1109/ICCV.2019.00140.

Lingxiao Huang and Nisheeth K. Vishnoi. Coresets for clustering in Euclidean spaces: importance sampling is nearly optimal. In *STOC*, pages 1416–1429. ACM, 2020. doi:10.1145/3357713.3384296.

Lingxiao Huang, Shaofeng H.-C. Jiang, and Jianing Lou. The power of uniform sampling for k-median. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 13933–13956. PMLR, 2023. URL https://proceedings.mlr.press/v202/huang23j.html.

Lingxiao Huang, Jian Li, and Xuan Wu. On optimal coreset construction for Euclidean (k, z)-clustering. In *STOC*, pages 1594–1604. ACM, 2024. doi:10.1145/3618260.3649707.

Wasim Huleihel, Arya Mazumdar, Muriel Médard, and Soumyabrata Pal. Same-cluster querying for overlapping clusters. In *NeurIPS*, pages 10485–10495, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/8a94ecfa54dcb88a2fa993bfa6388f9e-Abstract.html.

Christina Ilvento. Metric learning for individual fairness. In *FORC*, LIPIcs, pages 2:1–2:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.FORC.2020.2.

Rolf Klein, Rainer Penninger, Christian Sohler, and David P. Woodruff. Tolerant algorithms. In *ESA*, Lecture Notes in Computer Science, pages 736–747. Springer, 2011. doi:10.1007/978-3-642-23719-5_62.

Matthäus Kleindessner and Ulrike von Luxburg. Dimensionality estimation without distances. In *AISTATS*, JMLR Workshop and Conference Proceedings. JMLR.org, 2015. URL http://proceedings.mlr.press/v38/kleindessner15.html.

Matthäus Kleindessner and Ulrike von Luxburg. Kernel functions based on triplet comparisons. In *NeurIPS*, pages 6807–6817, 2017a. URL https://proceedings.neurips.cc/paper/2017/hash/07211688a0869d995947a8fb11b215d6-Abstract.html.

Matthäus Kleindessner and Ulrike von Luxburg. Lens depth function and k-relative neighborhood graph: Versatile tools for ordinal data analysis. *J. Mach. Learn. Res.*, 18:58:1–58:52, 2017b. URL https://jmlr.org/papers/v18/16-061.html.

Ken Lang. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Morgan Kaufmann, 1995. URL https://www.sciencedirect.com/science/article/pii/B9781558603776500487.

Marc T. Law, Nicolas Thome, and Matthieu Cord. Quadruplet-wise image similarity learning. In *ICCV*, pages 249–256. IEEE Computer Society, 2013. doi:10.1109/ICCV.2013.38.

Konstantin Makarychev, Yury Makarychev, and Ilya P. Razenshteyn. Performance of Johnson-Lindenstrauss transform for k-means and k-medians clustering. *SIAM J. Comput.*, 52(on): STOC19–269–STOC19–297, 2019. doi:10.1137/20M1330701.

Chris Meek, Bo Thiesson, and David Heckerman. US Census Data (1990). UCI Machine Learning Repository, 2001. URL https://archive.ics.uci.edu/ml/datasets/US+Census+Data+(1990). DOI: https://doi.org/10.24432/C5VP42.

Ramgopal R. Mettu and C. Greg Plaxton. Optimal time bounds for approximate clustering. *Machine Learning*, 56(1–3):35–60, 2004.

Aditya Narasimhan. *Combinatorial Algorithms in the Approximate Computing Paradigm*. PhD thesis, University of Oklahoma, Norman, Oklahoma, 2023. URL https://shareok.org/handle/11244/338857.

Shyam Narayanan and Jelani Nelson. Optimal terminal dimensionality reduction in Euclidean space. In *STOC*, pages 1064–1069. ACM, 2019.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

Pinki Pradhan, Anup Bhattacharya, and Ragesh Jaiswal. Improved algorithms for clustering with noisy distance oracles. In *The 29th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2026. URL https://openreview.net/forum?id=6lFHvZzieB.

Rahul Raychaudhury, Wen-Zhi Li, Syamantak Das, Sainyam Galhotra, and Stavros Sintos. Metric clustering and graph optimization problems using weak comparison oracles. In *COLT*, Proceedings of Machine Learning Research, pages 4777–4830. PMLR, 2025. URL https://proceedings.mlr.press/v291/raychaudhury25a.html.

Rahul Raychaudhury, Aryan Esmailpour, Sainyam Galhotra, and Stavros Sintos. Metric k-clustering using only weak comparison oracles. In *14th International Conference on Learning Representations, ICLR 2026*, 2026. URL https://openreview.net/forum?id=KmMEQOtXAy.

Chris Schwiegelshohn and Omar Ali Sheikh-Omar. An empirical evaluation of k-means coresets. In *ESA*, LIPIcs, pages 84:1–84:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ESA.2022.84.

Sandeep Silwal, Sara Ahmadian, Andrew Nystrom, Andrew McCallum, Deepak Ramachandran, and Seyed Mehran Kazemi. Kwikbucks: Correlation clustering with cheap-weak and expensive-strong signals. In *ICLR*. OpenReview.net, 2023. URL https://openreview.net/forum?id=p0JSSa1AuV.

Christian Sohler and David P. Woodruff. Strong coresets for k-median and subspace approximation: Goodbye dimension. In *FOCS*, pages 802–813. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00081.

Omer Tamuz, Ce Liu, Serge J. Belongie, Ohad Shamir, and Adam Kalai. Adaptively learning the crowd kernel. In *ICML*, pages 673–680. Omnipress, 2011. URL https://icml.cc/2011/papers/395_icmlpaper.pdf.

Antti Ukkonen. Crowdsourced correlation clustering with relative distance comparisons. In *ICDM*, pages 1117–1122. IEEE Computer Society, 2017. doi:10.1109/ICDM.2017.148.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *SIGIR*, pages 641–649. ACM, 2024. doi:10.1145/3626772.3657878.

Haike Xu, Sandeep Silwal, and Piotr Indyk. A bi-metric framework for fast similarity search. *CoRR*, abs/2406.02891, 2024. doi:10.48550/ARXIV.2406.02891.

A Deferred Proofs

This appendix collects the proofs deferred from the main text. Each subsection corresponds to a section of the body.

A.1 Proof from Section 2

Proof of Lemma 2.2. Let $F = \mathrm{cost}_z(C, P)$, $F_1 = \sum_{s \in S_1} w_1(s)\,\mathrm{dist}(s, C)^z$, $F_2 = \sum_{s \in S_2} w_2(s)\,\mathrm{dist}(s, C)^z$. Then $|F - F_1| \le \epsilon_1 F$ and $|F_1 - F_2| \le \epsilon_2 F_1 \le \epsilon_2(1+\epsilon_1)F$. By the triangle inequality: $|F - F_2| \le (\epsilon_1 + \epsilon_2 + \epsilon_1\epsilon_2)F$. □

A.2 Proofs from Section 4.1

Proof of Lemma 4.3. For any set of k centers $A \subseteq \mathcal{X}$ with $|A| \le k$, the triangle inequality (2) with $t = 1$ gives $\mathrm{dist}(\mathcal{M}(p), A)^z \le 2^{z-1}(\mathrm{dist}(p, \mathcal{M}(p))^z + \mathrm{dist}(p, A)^z)$. Summing over all $p \in P$, $\mathrm{cost}_z(A, C_w^0) \le 2^{z-1}(\alpha_0 \mathrm{OPT} + \mathrm{cost}_z(A, P))$. Taking A to be an optimal k-center set, $\min_{|A| \le k} \mathrm{cost}_z(A, C_w^0) \le 2^{z-1}(1+\alpha_0)\mathrm{OPT}$. Mettu–Plaxton (Lemma 4.2) yields $\mathrm{cost}_z(C, C_w^0) \le \beta_z \cdot 2^{z-1}(1+\alpha_0)\mathrm{OPT}$. Applying the triangle inequality (2) again with $t = 1$, $\mathrm{dist}(p, C)^z \le 2^{z-1}(\mathrm{dist}(p, \mathcal{M}(p))^z + \mathrm{dist}(\mathcal{M}(p), C)^z)$. Summing, $\mathrm{cost}_z(C, P) \le 2^{z-1}(\alpha_0 \mathrm{OPT} + \mathrm{cost}_z(C, C_w^0)) \le \big(2^{z-1}\alpha_0 + 4^{z-1}\beta_z(1+\alpha_0)\big)\mathrm{OPT}$. □

Proof of Lemma 4.4. Fix $p \in P_i$. By definition of the partition, c_i is the nearest center in C to $\mathcal{M}(p)$, so $\mathrm{dist}(\mathcal{M}(p), c_i) = \mathrm{dist}(\mathcal{M}(p), C)$. Applying the triangle inequality (2) with $t = 1$ first to the triple $(p, c_i, \mathcal{M}(p))$,

$$\mathrm{dist}(p, c_i)^z \le 2^{z-1}\big(\mathrm{dist}(p, \mathcal{M}(p))^z + \mathrm{dist}(\mathcal{M}(p), c_i)^z\big).$$

Then applying (2) again with $t = 1$ to $\mathrm{dist}(\mathcal{M}(p), c_i)^z = \mathrm{dist}(\mathcal{M}(p), C)^z$ via the triple $(\mathcal{M}(p), C, p)$,

$$\mathrm{dist}(\mathcal{M}(p), c_i)^z = \mathrm{dist}(\mathcal{M}(p), C)^z \le 2^{z-1}\big(\mathrm{dist}(\mathcal{M}(p), p)^z + \mathrm{dist}(p, C)^z\big).$$

Substituting, $\mathrm{dist}(p, c_i)^z \le (2^{z-1}+4^{z-1})\,\mathrm{dist}(p, \mathcal{M}(p))^z + 4^{z-1}\,\mathrm{dist}(p, C)^z$. Summing over all $p \in P_i$ and then over all i and applying Lemma 4.3, $\sum_i \mathrm{cost}_z(c_i, P_i) \le (2^{z-1}+4^{z-1})\sum_p \mathrm{dist}(p, \mathcal{M}(p))^z + 4^{z-1}\mathrm{cost}_z(C, P) \le (2^{z-1}+4^{z-1})\alpha_0\mathrm{OPT} + 4^{z-1}\alpha\mathrm{OPT}$. □

A.3 Proof from Section 4.3

Proof of Lemma 4.6. The farthest point in P_i has true rank n_i, so its position in π_i is no less than $n_i - D$, and thus it is among the queried positions. Whereas, $(d_{\max}^{(i)})^z = \max_p \mathrm{dist}(p, c_i)^z \le \sum_p \mathrm{dist}(p, c_i)^z = \mathrm{cost}_z(c_i, P_i)$. □

A.4 Proof from Section 4.4

Proof of Lemma 4.7. *Points in ambiguity zones* (Step 4.3, first case). Here $\mathrm{dist}(p, c_i)$ was queried in Steps 4.1–4.2, so classification is by direct comparison with thresholds.

Points outside all ambiguity zones (Step 4.3, second case). Point $p = \pi_i[j]$ satisfies $j_{\ell'-1} + D < j$ and $j < j_{\ell'} - D$ for some ℓ'. The dislocation bound gives p's true rank $r \in [j - D, j + D]$. The search guarantee gives the true boundary within D of each j_ℓ. Combining, r is past boundary $\ell'-1$ and before boundary ℓ', placing p in ring ℓ'.

Query count. Step 4.1 uses $O(\log n + D)$ per boundary. Step 4.2 uses $O(D)$ new queries per boundary. Total $O(L_i \cdot (\log n + D)) = O(L_i \log n)$. □

A.5 Proofs from Section 4.5

We first restate the iterative size-reduction lemma used in the proof of Theorem 4.8.

Lemma A.1 (Iterative size reduction; Lemma 7 of Cohen-Addad et al., 2021a)**.** *Let $q \in \mathbb{R}^d$ and let $R \subset \mathbb{R}^d$ be a ratio-2 ring around q. Suppose a uniform sample of size $s \in \widetilde{O}(\Gamma \cdot \log|R| \cdot \log(1/\delta))$ satisfies, with probability at least $1-\delta$, for all $A \subset \mathbb{R}^d$ with $|A| \le k$,*

$$\left|\mathrm{cost}_z(A,R) - \tfrac{|R|}{s}\mathrm{cost}_z(A,\Omega)\right| \le \epsilon\big(\mathrm{cost}_z(q,R) + \mathrm{cost}_z(A,R)\big).$$

Then $\widetilde{O}(\Gamma \cdot \log(1/\delta))$ uniform samples from R yield the same guarantee with probability at least $1-\delta$.

Proof of Theorem 4.8. Theorem 5.2 of Braverman et al. [2022] establishes that for a ratio-2 ring R around q, a uniform sample of size $2^{O(z\log z)} \cdot \widetilde{O}(\epsilon^{-2}) \cdot k \cdot \mathrm{sdim}(\mathcal{F}) \cdot \log(1/\delta)$ satisfies, with probability at least $1-\delta$, for all $A \subset \mathbb{R}^d$ with $|A| \le k$,

$$\left|\mathrm{cost}_z(A,R) - \tfrac{|R|}{s}\mathrm{cost}_z(A,\Omega)\right| \le \epsilon\big(\mathrm{cost}_z(q,R) + \mathrm{cost}_z(A,R)\big),$$

where $\mathrm{sdim}(\mathcal{F})$ denotes the shattering dimension of the ball range space in the ambient metric.

Case $d \le \epsilon^{-2}$. In $\mathbb{R}^d$, the shattering dimension of the ball range space is $\mathrm{sdim}(\mathcal{F}) = O(d)$, giving a sample of size $2^{O(z\log z)} \cdot \widetilde{O}(\epsilon^{-2}) \cdot k \cdot d \cdot \log(1/\delta)$.

Case $\epsilon^{-2} < d$. Since the cost function involves z-th powers of distances, we need the embedding to preserve $\|p-c\|^z$ up to a $(1\pm\epsilon)$ factor. Apply the Narayanan–Nelson terminal embedding [Narayanan and Nelson, 2019] with precision $\epsilon' = \epsilon/(2z)$, obtaining a map $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ with $d' = O(z^2\epsilon^{-2}\log|R|)$ satisfying $(1-\epsilon/(2z))\|p-c\| \le \|f(p)-f(c)\| \le (1+\epsilon/(2z))\|p-c\|$ for all $p \in R$ and $c \in \mathbb{R}^d$. Raising to the z-th power gives distortion $(1+\epsilon/(2z))^z \le 1+\epsilon$ on $\|\cdot\|^z$. In the embedded space, $\mathrm{sdim}(\mathcal{F}) = O(z^2\epsilon^{-2}\log|R|)$, giving a sample of size $2^{O(z\log z)} \cdot k \cdot \epsilon^{-4} \cdot \log|R| \cdot \mathrm{polylog}(k\epsilon^{-1}\log|R|/\delta)$, where the z^2 factor is absorbed into $2^{O(z\log z)}$. Applying Lemma A.1 removes the $\log|R|$ factor, yielding $2^{O(z\log z)} \cdot k \cdot \widetilde{O}(\epsilon^{-4}) \cdot \log(1/\delta)$.

The theorem is obtained by combining both cases. □

Proof of Lemma 4.9. With high probability π is approximately sorted with maximum displacement D, and by Lemma 4.7 every point in P is correctly assigned.

Fix $A \subset \mathbb{R}^d$ with $|A| \le k$. We bound $|\mathrm{cost}_z(A,P) - \mathrm{cost}_z(A,S_1)|$ by summing per-cluster contributions from the rings and the close set.

Ring error. By a union bound over all rings, each succeeding with probability at least $1 - n^{-c}/(\sum_{i=1}^k L_i)$, Theorem 4.8 with precision ϵ_1 gives that with high probability, simultaneously for all rings and all $|A| \le k$,

$$\begin{aligned}\sum_{i,\ell}\left|\mathrm{cost}_z(A,R_{i,\ell}) - \tfrac{|R_{i,\ell}|}{s}\mathrm{cost}_z(A,\Omega_{i,\ell})\right| &\le \epsilon_1 \sum_{i,\ell}\big(\mathrm{cost}_z(c_i,R_{i,\ell}) + \mathrm{cost}_z(A,R_{i,\ell})\big)\\ &\le \epsilon_1\big(\alpha'\mathrm{OPT} + \mathrm{cost}_z(A,P)\big),\end{aligned}$$

where the last inequality uses $\sum_{i,\ell}\mathrm{cost}_z(c_i,R_{i,\ell}) \le \sum_i \mathrm{cost}_z(c_i,P_i) \le \alpha'\mathrm{OPT}$ (Lemma 4.4; the rings partition a subset of each P_i) and $\sum_{i,\ell}\mathrm{cost}_z(A,R_{i,\ell}) \le \mathrm{cost}_z(A,P)$.

Close error. For each $p \in P_i^{\mathrm{cl}}$, the triangle inequality (3) gives

$$|\,\mathrm{dist}(p,A)^z - \mathrm{dist}(c_i,A)^z| \le \epsilon_1\ \mathrm{dist}(c_i,A)^z + (3z/\epsilon_1)^{z-1}\ \mathrm{dist}(p,c_i)^z.$$

Summing the second term over P_i^{cl},

$$(3z/\epsilon_1)^{z-1}\mathrm{cost}_z(c_i,P_i^{\mathrm{cl}}) \le (3z/\epsilon_1)^{z-1}\cdot n_i \cdot \tau_{\mathrm{cl}}^{(i)} \le \frac{\epsilon_1}{\alpha'}\cdot(d_{\max}^{(i)})^z \le \frac{\epsilon_1}{\alpha'}\mathrm{cost}_z(c_i,P_i),$$

using the definition of $\tau_{\mathrm{cl}}^{(i)}$ and Lemma 4.6. Summing the first term over P_i^{cl}, $\epsilon_1|P_i^{\mathrm{cl}}|\ \mathrm{dist}(c_i,A)^z \le \epsilon_1\, n_i\ \mathrm{dist}(c_i,A)^z$. To bound $\mathrm{dist}(c_i,A)^z$, we average the triangle inequality (2) over P_i.

Claim A.2. $\mathrm{dist}(c_i,A)^z \le 2^{z-1}\big(\mathrm{cost}_z(c_i,P_i) + \mathrm{cost}_z(A,P_i)\big)/n_i$.

Proof. Let $c(p) = \arg\min_{c \in A} \operatorname{dist}(p, c)$ for $p \in P_i$. Then $\operatorname{dist}(c_i, A)^z \le \operatorname{dist}(c_i, c(p))^z \le 2^{z-1}(\operatorname{dist}(c_i, p)^z + \operatorname{dist}(p, A)^z)$. Average over all $p \in P_i$. □

Hence $\epsilon_1 n_i \operatorname{dist}(c_i, A)^z \le 2^{z-1}\epsilon_1\big(\operatorname{cost}_z(c_i, P_i) + \operatorname{cost}_z(A, P_i)\big)$. Combining both terms, the total close error for cluster i is

$$E_i^{\mathrm{cl}} \le \big(\frac{\epsilon_1}{\alpha'} + 2^{z-1}\epsilon_1\big)\operatorname{cost}_z(c_i, P_i) + 2^{z-1}\epsilon_1 \operatorname{cost}_z(A, P_i).$$

Summing the close errors over all partitions,

$$\begin{aligned}\sum_{i=1}^{k} E_i^{\mathrm{cl}} &\le \big(\frac{\epsilon_1}{\alpha'} + 2^{z-1}\epsilon_1\big)\sum_{i=1}^{k}\operatorname{cost}_z(c_i, P_i) + 2^{z-1}\epsilon_1 \sum_{i=1}^{k}\operatorname{cost}_z(A, P_i)\\ &\le \big(\frac{\epsilon_1}{\alpha'} + 2^{z-1}\epsilon_1\big)\alpha'\mathrm{OPT} + 2^{z-1}\epsilon_1 \operatorname{cost}_z(A, P),\end{aligned}$$

where the last inequality follows from Lemma 4.4. Summing over the ring and close errors and applying the triangle inequality,

$$\begin{aligned}|\operatorname{cost}_z(A, P) - \operatorname{cost}_z(A, S_1)| &\le \epsilon_1\big(\alpha'\mathrm{OPT} + \operatorname{cost}_z(A, P)\big) + \big(\frac{\epsilon_1}{\alpha'} + 2^{z-1}\epsilon_1\big)\alpha'\mathrm{OPT} + 2^{z-1}\epsilon_1 \operatorname{cost}_z(A, P)\\ &\le \big(\alpha' + 1 + \epsilon_1 + 2^{z-1}\alpha' + 2^{z-1}\big)\epsilon_1 \operatorname{cost}_z(A, P) \le \frac{\epsilon}{3}\operatorname{cost}_z(A, P)\end{aligned}$$

using $\operatorname{cost}_z(A, P) \ge \mathrm{OPT}$ and $\epsilon_1 = \epsilon/(6 \cdot 2^z \cdot (1+\alpha'))$. □

A.6 Proof from Section 4.6

Proof of Lemma 4.11. Since S_1 is an $(\epsilon/3)$-coreset for P (Lemma 4.9), C remains $O(\alpha)$-approximate for S_1. Theorem 4.10 then applies and produces an $(\epsilon/3)$-coreset S_2 for S_1 of the stated size. □

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and Section 1 state the contributions, namely the first ϵ-coreset construction for (k, z)-clustering in the RM-model with size matching the classical bound and a polylogarithmic number of distance queries (Theorem 1.1), and the empirical evaluation in Section 5. The assumptions and parameter dependencies are explicit in the theorem statement.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The model assumptions and parameter dependencies are stated explicitly in Theorem 1.1, including the noise rate $\varphi < 1/4$, the restriction to Euclidean space, and the dependencies on z, k, d, and ϵ.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: All theorems and lemmas are numbered, and the assumptions are stated within or immediately before each statement.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 5 describes the datasets, encoders, parameters (k, φ, coreset sizes, sweep ranges), the implementation adaptations from the theoretical pipeline, and the evaluation protocol. Source code is to be made available at the URL listed in the paper.

   Guidelines:

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

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

5. **Open access to data and code**

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

   Answer: [Yes]

   Justification: All datasets used (20-Newsgroups, STL-10, Census 1990) are either publicly available or published in the source code repository provided in Section 5.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5 specifies the dataset sizes, the values of k and φ used in each experiment, the coreset sizes, the encoders (and their dimensions and parameter counts), the number of repetitions, and the worst-case cost ratio metric.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Each experimental cell is averaged over 10 statistically independent repetitions, and the figures show shaded standard-deviation bands across reps, as stated in Section 5.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Section 5 reports that experiments were run on AMD EPYC 9354 nodes (32 physical cores per socket, 502 GB RAM) running AlmaLinux 8.10 with Python.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research uses only publicly available datasets and pretrained models for benchmark evaluation of a theoretical algorithm, and does not involve human subjects or sensitive data.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: The paper presents a foundational theoretical result on coreset construction in an oracle access model. It is not tied to a specific application or deployment and has no direct path to negative societal impacts.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper does not release any new data or models that pose a risk of misuse.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All datasets and pretrained encoders used in the experiments are cited in Section 5 and Table 2, and are publicly available under their respective licenses (UCI ML Repository for Census 1990; the original publications for 20-Newsgroups and STL-10; Hugging Face and PyTorch Hub for the BGE, MobileNetV3, and DINOv2 encoders).

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [N/A]

Justification: The paper does not release any new datasets or pretrained models; the algorithmic implementation will be released as code, with documentation, at the URL given in Section 5.

Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [N/A]

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

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core methodology of this paper does not involve LLMs as a component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.