Fault Tolerant Coresets

Milind Prabhu
University of Michigan
Ann Arbor, MI, USA
milindpr@umich.edu

Chris Schwiegelshohn
Aarhus University
Aarhus, Denmark
schwiegelshohn@cs.au.dk

Sudarshan Shyam
King's College London
London, UK
sudarshan.shy@gmail.com

Abstract

Coresets are small weighted summaries of large datasets that help quickly evaluate some loss objectives on the input dataset. We mainly focus on coresets for the k-means objective, where the cost of the input dataset with respect to a set of k centers is the sum of squared distances from each point to its closest center. A k-means ε-coreset Ω satisfies $\text{cost}(P, S) = (1 \pm \varepsilon)\text{cost}(\Omega, S)$ for every set S of k centers, where P denotes the input dataset. A long line of work has established near optimal ε-coreset size bounds for k-means. A key assumption in these works is that the coreset remains intact after construction. The question we ask is the following: if an adversary corrupts at most f weighted entries of an ε-coreset, can we recover an $O(\varepsilon)$-coreset via a decoding algorithm that is given only the corrupted coreset. This models situations where entries of the coreset can be corrupted during storage or transmission, and recovery must proceed without access to the original dataset. Our main result shows that increasing the usual coreset size by an *additive* $O(fk/\varepsilon)$ yields a coreset that can be decoded into an $O(\varepsilon)$-coreset after any f adversarial corruptions. This bound is achieved by sensitivity sampling, a widely used method for constructing coresets. Our result is optimal in the sense that any fault tolerant coreset, whose decoding algorithm only keeps a subset of the points must also have size $\Omega(fk/\varepsilon)$. Our experiments demonstrate recovery under adversarial corruptions.

1 Introduction

Modern datasets are often too large to store, process, or analyze directly. Clustering is a basic tool for extracting structure from such data, and one of the most widely used clustering objectives is k-means. Given a point set $P \subseteq X$ in a metric space (X, d), the goal is to choose a set $S \subseteq X$ of k centers minimizing

$$\text{cost}(P, S) := \sum_{p \in P} \min_{s \in S} d(p, s)^2.$$

Over the last two decades, *coresets* have emerged as a successful tool for addressing this challenge. At a high level, a coreset is a small weighted summary of the input that approximates the objective value of every feasible solution. Formally, given an input point set P, a weighted set Ω is an ε-coreset if

$$\text{cost}(\Omega, S) \in (1 \pm \varepsilon)\text{cost}(P, S) \qquad \text{for every set } S \text{ of } k \text{ centers.}$$

Thus, one can run downstream clustering algorithms on Ω rather than on the full input P, while preserving guarantees for the original instance. Following a long line of work, we now have very small coreset bounds for k-means in finite n-point metrics ($\tilde{O}(k\varepsilon^{-2}\log n)$ Feldman and Langberg [2011]), metrics with bounded doubling dimension ($\tilde{O}(k\varepsilon^{-2}D)$ for doubling dimension D Cohen-Addad et al. [2021c]), and Euclidean k-means ($\tilde{O}(k\varepsilon^{-2}\min(\sqrt{k}, \varepsilon^{-2}))$ Cohen-Addad et al. [2022d], Huang et al. [2024]). All of these bounds are optimal up to polylogarithmic factors.

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

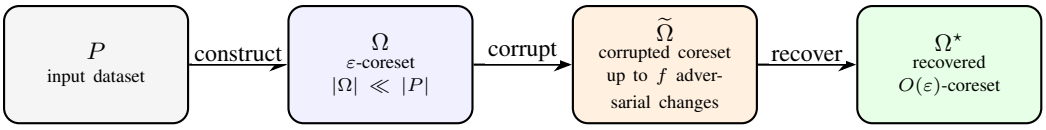


Figure 1: Fault-tolerant coreset pipeline. From the input dataset P, we construct a coreset Ω. An adversary then applies up to f corruptions, producing $\widetilde{\Omega}$. The goal is to recover from $\widetilde{\Omega}$ a set $\Omega^\star$ that is again a $O(\varepsilon)$-coreset for the original dataset P.

These results assume that the coreset is accessed exactly as constructed. In many settings, this assumption may be too strong. A coreset may be stored for long periods of time, transmitted across unreliable channels, or passed through complex software systems. In each case, some of its entries may be altered due to hardware faults, packet loss, software errors, or adversarial attacks. This motivates the central question of this work.

> Can coreset compression be made tolerant to adversarial corruptions?

We study this question in a post-compression corruption model. First, a coreset Ω is constructed from the input P. Then an adversary may corrupt up to f weighted entries of Ω. Corrupting an entry means that both its location and its weight may be changed arbitrarily. The decoder is given only the corrupted summary $\widetilde{\Omega}$, not the original input P, and must recover from it a valid coreset for P; see Figure 1. Corruption models for samples drawn from a distribution have a long history in robust statistics, beginning with the seminal works Tukey [1960], Huber [1964], and continuing through later work on adversarial contamination in high-dimensional statistics (see Section 1.3).

The above question exposes a basic tension. The purpose of a coreset is to compress: it achieves small size by discarding most of the redundant information in the input. Fault tolerance, on the other hand, seems to require redundancy, since the summary must remain useful even after some of its entries are corrupted. Thus, it is intriguing whether one can protect a coreset against corruptions without giving up the compression that makes it useful.

1.1 Our Contributions

We answer the main question of this paper in the affirmative for a broad class of pointwise-loss objectives, including k-means. Our main conceptual contribution is that sensitivity sampling, one of the most general and widely used frameworks for coreset construction, can be made tolerant to post-compression adversarial corruptions.

We first discuss the sensitivity sampling framework in its general form. Formally, we consider objectives that decompose as sums of pointwise losses: there is a solution space $\mathcal{S}$, and for every input point $p \in P$, there is a nonnegative loss function $\ell_p : \mathcal{S} \to \mathbb{R}_{\geq 0}$ such that $L_P(S) = \sum_{p \in P} \ell_p(S)$ for every $S \in \mathcal{S}$. The k-means objective is a central example: here $\mathcal{S}$ is the set of all k-point subsets in the metric space in consideration, and $\ell_p(S) = \min_{s \in S} d(p, S)^2$.

The sensitivity of a point $p \in P$ is

$$\sigma_P(p) := \sup_{S \in \mathcal{S} \,:\, L_P(S) > 0} \frac{\ell_p(S)}{L_P(S)},$$

and the total sensitivity is $T := \sum_{p \in P} \sigma_P(p)$. Sensitivity sampling constructs a coreset by sampling points with probability proportionate to upper bounds on their sensitivities and reweighting the sampled points appropriately. This framework is known to yield near-optimal coreset size bounds for a variety of problems, including the aforementioned clustering results, but also linear regression Woodruff and Yasuda [2024], logistic regression Munteanu and Omlor [2024], low-rank approximation Cohen et al. [2017], and spectral sparsification Spielman and Srivastava [2011].

Our first result shows that the same framework can be made fault-tolerant. If a standard sensitivity-sampling construction has total sensitivity T and produces an ε-coreset of size m, then adding $O(fT/\varepsilon)$ samples suffices to tolerate f adversarial corruptions of the coreset.

Theorem 1.1 (Fault-tolerant sensitivity sampling)**.** *Fix $\varepsilon \in (0, 1/2)$ and $f \in \mathbb{N}$. Consider any pointwise-loss objective admitting a sensitivity-sampling ε-coreset of size m and total sensitivity T. Then there is an f-fault-tolerant $O(\varepsilon)$-coreset of size*

$$m + O(fT/\varepsilon).$$

For a constant number of faults, the additive overhead is typically small compared with standard sensitivity-sampling coreset sizes, which in many cases are of order $\Omega(T/\varepsilon^2)$. Nevertheless, one could ask if the $O(fT/\varepsilon)$ term could be improved. It turns out that this is not possible, even for k-means. Consider that for k-means, the total sensitivity is $T = \Theta(k)$ Langberg and Schulman [2010] and thus, the additive overhead in Theorem 1.1 becomes $O(fk/\varepsilon)$. The following theorem shows that this dependence on f, k, and ε is necessary in the worst case for coresets whose decoding algorithm only keeps a subset of the points with their original weights.

Theorem 1.2 (Lower bound for fault-tolerant k-means coresets)**.** *For every $\varepsilon \in (0, 1/2)$ and $f \in \mathbb{N}$, there are k-means instances for which any f-fault-tolerant ε-coreset supported on the input points, whose decoding algorithm only keeps a subset of its entries with their original weights, must have size $\Omega(fk/\varepsilon)$.*

Fault tolerance is not only an existential question: ideally we would like an efficient decoding algorithm that can recover a valid coreset from the corrupted summary. In our framework, the decoder relies on identifying points with unusually large sensitivity. Consequently, the effectiveness of the decoder depends on computing, or accurately approximating, sensitivities on the corrupted coreset.

While the sensitivities can be computed exactly up to numerical precision for some problems, this is a substantial challenge for k-means. Existing sensitivity-sampling implementations typically use bicriteria solutions or other proxy upper bounds instead Feldman and Langberg [2011], Braverman et al. [2016], Bansal et al. [2024]. These upper bounds suffice for sampling, but may substantially overestimate individual sensitivities and therefore need not identify which points to remove. Here, we relate the approximability of the k-means sensitivity of a point to the approximability of (constrained) k-means.

Theorem 1.3 (Approximability of k-means sensitivities)**.** *In finite metrics, pointwise sensitivity approximation for k-means is no harder than constrained weighted k-means with admissible center sets, and is at least as hard as finite-metric k-means up to arbitrarily small constant slack.*

In other words, we show that any α-approximation algorithm can be used to compute an α-approximation to the sensitivity of a point p (Theorem 4.1). Conversely, if it is NP-hard to approximate k-means beyond a factor α, then approximating the sensitivity beyond the factor α is likewise NP-hard (Theorem 4.2). We review available algorithms in more detail in Appendix B.

Experiments. We conduct three sets of experiments. First, we test recovery under Gaussian and adversarial corruptions. In an exact finite-domain adversarial test on Adult's age coordinate ($k = 2$, $m = 1000$, $f = 5$), decoding reduces mean distortion from 46.875 to 1.072. Second, on Taxi ($k = 50$, $f = 5$), sensitivity sampling achieves mean witnessed distortion 1.124, compared with 1.689 for BICO Fichtenberger et al. [2013] and 1.686 for StreamKM++ Ackermann et al. [2012]. Finally, on CoverType ($k = 2$, $m = 100$), sampling with exact finite-domain sensitivities achieves mean empirical distortion 1.080, versus 1.115 for the chosen proxy. A compression-regime simplex example further separates these distributions even with an optimal reference partition.

1.2 Technical Overview

As a first observation, any coreset can be made tolerant to f corruptions by storing $2f + 1$ labeled copies of each coreset entry and recovering each entry by majority vote. This protects against f corruptions, but multiplies the coreset size by $2f + 1$. Our goal is to avoid this multiplicative blow-up and obtain fault tolerance with only an additive increase in size.

In what follows, we explain the main ideas using the 1-means objective on the line. The same principle underlies the construction for more general cost functions. In 1-means on the line, a solution is a single center c, and the cost of a weighted set Q is

$$\text{cost}(Q, c) = \sum_{q \in Q} w(q) d(q, c)^2.$$

A non-sampling coreset for 1-means can be extremely compact: in fact, one can construct a coreset with only two weighted points that preserves the mean and variance Feldman et al. [2020], and therefore preserves the cost of every center. Such a summary is accurate when accessed exactly, but a corruption of one of its two weighted entries can substantially change the objective it represents.

Coresets that concentrate substantial weight on a few representatives can be vulnerable to corruption. Corrupting one high-weight summary point can significantly change the represented objective. Our construction uses specific properties of sensitivity sampling to avoid a multiplicative overhead in the coreset size.

Why sensitivity sampling helps. Sensitivity sampling behaves differently: it spreads importance across sampled points. Once the sampled points have sufficiently uniform importance, the cost of every fixed solution is preserved by concentration Langberg and Schulman [2010], Feldman and Langberg [2011]. For 1-means, after drawing a coreset of size m, every uncorrupted sampled point has weighted sensitivity on the order of $1/m$ with high probability; for k-means, the corresponding scale is $O(k/m)$.

This pointwise control is the source of fault tolerance. Consider first a single corruption. If the corrupted point has much larger sensitivity than the typical scale, then it is detectable and can be removed. If its sensitivity is comparable to the typical scale, then it may be impossible to identify, but its effect on every candidate solution is already small. Thus, a corruption is either detectable or harmless.

A Sensitivity-Based Iterative Decoder. For multiple corruptions, one might try to compute all sensitivities once and remove the f largest-sensitivity points, an idea used in outlier-resistant coreset constructions Feldman and Langberg [2011], In et al. [2025] and in certain deterministic coreset constructions McCurdy [2018], Papailiopoulos et al. [2014]. This fails because corruptions can mask one another. An example already exists for the 1-means problem: Suppose the uncorrupted coreset of size m lies at the origin and the adversary corrupts two entries. They move one point p_2 to 1 and another point p_1 to a very large positive value. For any solution, the contribution of p_2 to the cost will be strictly less than $\frac{1}{m-1}$, whereas the sensitivity of p_1 is close to 1 and the sensitivity of the remaining $m-2$ points at the origin is at least $\frac{1}{m-1}$. A one-shot pruning rule according to sensitivities will always retain p_2 and the adversary successfully corrupts the coreset.

Our decoder therefore uses an iterative peeling approach. Starting from the corrupted coreset, it recomputes sensitivities with respect to the current weighted set and removes a point of maximum sensitivity, repeating for f rounds. Notice how in the aforementioned example the sensitivity of p_2 immediately becomes 1 after p_1 is removed. It turns out this must be the case in general: At each step, either a high-influence corruption is exposed and removed, or all remaining corruptions have small total effect on the cost of any solution. While the exposition here mainly focused on the 1-means objective where sensitivities can be computed exactly, this is not the case for k-means. Nevertheless, the same arguments also work when using approximate sensitivities for each point.

Further comments and limitations. Our framework applies to problems that admit coresets via sensitivity sampling. It does not immediately apply to settings where sensitivity sampling is not the right coreset mechanism, such as certain fair clustering objectives Schmidt et al. [2019], or to geometric approximation problems such as directional width Agarwal et al. [2004]. Whether fault-tolerant coreset constructions with better than a multiplicative $O(f)$ overhead exist for these problems remains open.

1.3 Related Work

Adversarial contamination and robust statistics. Our corruption model is conceptually related to adversarial contamination models in robust statistics and learning, where a bounded fraction of samples may be changed arbitrarily. Such models have a long history, beginning with the classical works of Tukey [1960], Huber [1964], and have recently received significant attention in high-dimensional statistics and theoretical machine learning Diakonikolas et al. [2016], Lai et al. [2016], Blanc et al. [2022], Canonne et al. [2023], Blanc and Valiant [2025]. This line of work studies robust versions of basic statistical tasks, including mean and covariance estimation, regression, stochastic

optimization, and mixture learning; see also the book of Diakonikolas and Kane [2023] and the survey of Lugosi and Mendelson [2019].

Coresets. Coresets are a central tool for compressing large optimization instances while approximately preserving the objective value of every feasible solution. The earliest examples focused primarily on geometric objectives Agarwal et al. [2005]. The connection to learning theory was established in a series of papers Chen [2009], Langberg and Schulman [2010], Feldman and Langberg [2011] and has been since studied for a variety of problems including clustering Bansal et al. [2024], Becchetti et al. [2019], Braverman et al. [2022, 2021], Cohen-Addad et al. [2025, 2022c, 2023], Draganov et al. [2024], Har-Peled and Mazumdar [2004], Huang et al. [2018], Huang and Vishnoi [2020], regression Boutsidis et al. [2013], Dasgupta et al. [2009], Mai et al. [2021], Munteanu and Omlor [2024], Munteanu et al. [2018], Woodruff and Yasuda [2024], low-rank approximation Cohen et al. [2017], Feldman et al. [2013], Feng et al. [2021], Maalouf et al. [2020], Sohler and Woodruff [2018], Varadarajan and Xiao [2012], and spectral sparsification Bansal et al. [2019], Jambulapati et al. [2023], Lee and Sun [2018], Lee [2023], Spielman and Srivastava [2011]. Many of these results are based on sensitivity sampling, sometimes also known as leverage score sampling, or, in particular in the context of graph sparsification, sampling by electrical resistances. Indeed, for (variants of) these aforementioned problems, sensitivity sampling yields coresets that are optimal up to logarithmic factors. We refer to surveys for more pointers, see Feldman [2020].

Somewhat related are robust coresets and coresets for outliers Feldman and Langberg [2011], Huang et al. [2018, 2023, 2025a], In et al. [2025]. Here, the cost of a query on the coreset after removing the most expensive points must remain the same. Nevertheless, as mentioned in the technical overview, there are differences and these methods do not seem to imply results for fault-tolerant coresets. The closest related work in spirit is the recent paper of Huang et al. [2025b], which studies coresets for clustering under stochastic noise. There, the input dataset is corrupted by stochastic noise before the coreset is constructed, and the coreset is built from the noisy input. In contrast, in our model the coreset is first realized and then an adversary, with full knowledge of the realized coreset, corrupts entries of the coreset itself.

2 Model and Preliminaries

We use the following notation throughout. A pointwise-loss objective consists of an input set P, a solution space $\mathcal{S}$, and nonnegative loss functions $\{\ell_p : \mathcal{S} \to \mathbb{R}_{\geq 0}\}_{p \in P}$. For a solution $S \in \mathcal{S}$, write $L_P(S) := \sum_{p \in P} \ell_p(S)$. For a weighted multiset Q with weights w_Q, we define $\ell_q^w(S) := \ell_q(S) \cdot w_q$ and analogously $L_Q^w(S) := \sum_{q \in Q} \ell_q^w(S)$. For k-means in a metric space (X, d), $\mathcal{S}$ is the collection of k-point subsets of X and $\ell_p(S) := d(p, S)^2 := \min_{s \in S} d(p, s)^2$.

A weighted multiset Ω is an ε-coreset for P if $L_\Omega(S) \in (1 \pm \varepsilon) L_P(S)$ for every $S \in \mathcal{S}$. Note that most coresets require using weighted losses.

The Corruption Model and Fault-Tolerant Coresets An adversary corrupts an entry $(q, w_\Omega(q))$ of a weighted multiset Ω by replacing it with an arbitrary weighted entry (q', w'), where q' lies in the ambient domain and $w' \geq 0$. For an integer $f \geq 0$, an f-fault corruption of Ω is any weighted multiset $\widetilde{\Omega}$ obtained by corrupting at most f entries of Ω.

We now define what it means for a coreset to remain recoverable in this corruption model.

Definition 2.1 (f-fault-tolerant coreset)**.** Fix $\varepsilon \in (0, 1)$. A pair $(\Omega, \textsc{Dec})$ is an *f-fault-tolerant ε-coreset* for P if, with constant probability over the construction of Ω, the following event holds:

(i) Ω is an ε-coreset for P;

(ii) for every f-fault corruption $\widetilde{\Omega}$ of Ω, the decoder returns a weighted submultiset $\Omega^\star = \textsc{Dec}(\widetilde{\Omega}) \subseteq \widetilde{\Omega}$, with inherited weights, that is an ε-coreset for P.

Conditioned on the success event for the initial coreset, the guarantee must hold for every admissible adversarial corruption.

3 Fault-Tolerant Coreset Framework

In this section, we give a general fault-tolerant coreset construction for objectives that admit sensitivity sampling. The initial summary is sampled by standard sensitivity sampling: each point is sampled with probability proportional to its sensitivity and assigned the corresponding inverse-probability weight. The decoder uses approximate sensitivities only to identify points to filter.

Let T denote the total sensitivity of P, and let $m_{\mathrm{sc}}(\varepsilon)$ denote the number of samples required for standard sensitivity sampling to return an ε-coreset for P with constant probability.

For a weighted multiset Q with weights w, write $\mathrm{cost}(Q,S) := \sum_{q\in Q} w(q)\ell_q(S)$. The sensitivity of an entry $q \in Q$ is

$$\sigma_Q(q) := \sup_{S\in\mathcal{S}:\,\mathrm{cost}(Q,S)>0} \frac{w(q)\ell_q(S)}{\mathrm{cost}(Q,S)}.$$

For $\alpha \geq 1$, an α-approximate sensitivity of a point q in a weighted multiset Q is a value $\widetilde{\sigma}_Q(q)$ satisfying

$$\frac{1}{\alpha}\sigma_Q(q) \leq \widetilde{\sigma}_Q(q) \leq \sigma_Q(q).$$

Assume that, for some constant $\alpha \geq 1$, α-approximate sensitivities can be computed efficiently for any point in any weighted set.

Theorem 3.1 (Fault-tolerant coreset)**.** *For any $\varepsilon \in (0, 1/10)$ and $f \in \mathbb{N}$, sensitivity sampling from P with*

$$m = m_{\mathrm{sc}}(\varepsilon) + \frac{\alpha f T}{\varepsilon}$$

samples admits an efficient filtering decoder that recovers a 6ε-coreset from any f-fault corruption with constant probability over the construction of Ω.

The decoder is simple: it repeatedly recomputes approximate sensitivities on the current corrupted summary and deletes a point of maximum approximate sensitivity. The proof uses two facts. First, sensitivity sampling ensures that every sampled copy has small weighted contribution (Lemma 3.2). Second, if the decoder ever deletes an uncorrupted point, then all remaining corrupted points must have small sensitivity and hence small total cost.

Lemma 3.2. *Let Ω be obtained by sensitivity sampling m points from P. Then, for every sampled copy $x \in \Omega$ and every solution S,*

$$w(x)\mathrm{cost}(x,S) \leq \frac{T}{m}\mathrm{cost}(P,S).$$

Proof. Let σ denote the sensitivities and let $T := \sum_{p\in P}\sigma(p)$. Sensitivity sampling chooses point p with probability $q_p = \sigma(p)/T$ and assigns weight $1/(mq_p)$.

For a sampled copy x, we have $w(x) = T/(m\sigma(x))$. By the definition of sensitivity,

$$\mathrm{cost}(x,S) \leq \sigma(x)\mathrm{cost}(P,S)$$

for every solution S. Hence

$$w(x)\mathrm{cost}(x,S) = \frac{T}{m\sigma(x)}\mathrm{cost}(x,S) \leq \frac{T}{m}\mathrm{cost}(P,S). \qquad \square.$$

Algorithm 1 DEC$(\widetilde{\Omega}, f)$

Require: Corrupted weighted multiset $\widetilde{\Omega}$ and budget f
Ensure: Weighted multiset $\Omega^\star$

```
Q_0 ← Ω̃
for i = 1, ..., f do
    Compute an α-approximation σ̃_{Q_{i-1}}(x) to σ_{Q_{i-1}}(x) for each x ∈ Q_{i-1}
    Choose x_i ∈ arg max_{x∈Q_{i-1}} σ̃_{Q_{i-1}}(x)
    Q_i ← Q_{i-1} \ {x_i}
end for
return Ω* ← Q_f
```

Proof of Theorem 3.1. Let Q_i denote the weighted multiset remaining after i deletions when Algorithm 1 is run on an arbitrary f-fault corruption $\widetilde{\Omega}$, so $Q_0 = \widetilde{\Omega}$. The algorithm outputs Q_f. We will show that Q_f is a 6ε-coreset for P. Condition on Ω being an ε-coreset, an event of constant probability, and write $Q_i = G_i \cup B_i$ for its uncorrupted and corrupted entries.

Lower bound. We show that $\mathrm{cost}(Q_f, S) \geq (1-6\varepsilon)\mathrm{cost}(P, S)$ for every solution S. Each G_i omits at most f corrupted samples and at most $i \leq f$ samples deleted by the decoder. By Lemma 3.2 and $m \geq \alpha f T/\varepsilon$, the total lost cost is at most $2fT\mathrm{cost}(P, S)/m \leq 2\varepsilon\mathrm{cost}(P, S)$. Since $G_i \subseteq \Omega$, we obtain, simultaneously for every solution S and $0 \leq i \leq f$,

$$(1-3\varepsilon)\mathrm{cost}(P, S) \leq \mathrm{cost}(G_i, S) \leq (1+\varepsilon)\mathrm{cost}(P, S). \tag{1}$$

Since Q_f contains G_f and costs are nonnegative,

$$\mathrm{cost}(Q_f, S) \geq \mathrm{cost}(G_f, S) \geq (1-3\varepsilon)\mathrm{cost}(P, S) \geq (1-6\varepsilon)\mathrm{cost}(P, S).$$

Upper bound. We show that $\mathrm{cost}(Q_f, S) \leq (1+6\varepsilon)\mathrm{cost}(P, S)$ for every solution S. If $B_f = \emptyset$, then $Q_f = G_f$ and the bound follows directly from (1). Otherwise, since the decoder makes f deletions and at most f entries were corrupted, it must delete an uncorrupted point. Let g be the first such point, and let $Q = G \cup B$ be the summary just before its deletion. This deletion lets us bound the sensitivity of every corruption still present. By Lemma 3.2, (1), and $Q \supseteq G$,

$$\sigma_Q(g) \leq \frac{T}{(1-3\varepsilon)m} \leq \frac{2\varepsilon}{\alpha f},$$

where we used $\varepsilon < 1/10$. Since g maximizes the approximate sensitivity, every $b \in B$ satisfies

$$\sigma_Q(b) \leq \alpha\widetilde{\sigma}_Q(b) \leq \alpha\widetilde{\sigma}_Q(g) \leq \alpha\sigma_Q(g) \leq \frac{2\varepsilon}{f}.$$

Summing over $|B| \leq f$ gives $\mathrm{cost}(B, S) \leq 2\varepsilon\mathrm{cost}(Q, S)$ for every S. Thus,

$$(1-2\varepsilon)\mathrm{cost}(Q, S) \leq \mathrm{cost}(G, S) \leq (1+\varepsilon)\mathrm{cost}(P, S).$$

The remaining deletions can only decrease costs, so

$$\mathrm{cost}(Q_f, S) \leq \mathrm{cost}(Q, S) \leq \frac{1+\varepsilon}{1-2\varepsilon}\mathrm{cost}(P, S) \leq (1+6\varepsilon)\mathrm{cost}(P, S),$$

where the last inequality uses $\varepsilon < 1/10$. Together with the lower bound, this proves that Q_f is a 6ε-coreset. □

Remark 3.3 (Proxy sensitivity upper bounds). Theorem 3.1 is stated for exact sensitivity sampling in the initial construction. More generally, the same framework applies when the initial sampling distribution is defined by proxy scores $\tau(p)$ satisfying $\tau(p) \geq \sigma_P(p)$ for every $p \in P$. Such sensitivity upper bounds are common in many coreset constructions, since they are often easier to compute than exact sensitivities and can simplify the analysis. See Section A.1 for details.

Tightness for k-means. The additive overhead in Theorem 3.1 is tight for k-means. The lower bound holds even for the simple Euclidean instance $P = \{e_1, \ldots, e_k\} \subset \mathbb{R}^k$ and for filtering decoders whose output is a submultiset of the corrupted coreset with inherited weights.

Theorem 3.4 (Lower bound for fault-tolerant k-means coresets)**.** *Fix $\varepsilon \in (0, 1/2)$ and $f \in \mathbb{N}$. Let $P = \{e_1, \ldots, e_k\} \subset \mathbb{R}^k$. Any f-fault-tolerant ε-coreset (Ω, DEC) for Euclidean k-means on P, where Ω is a weighted multiset supported on P and* DEC *only filters points with inherited weights, must satisfy $|\Omega| = \Omega(fk/\varepsilon)$.*

The proof uses the fact that any coreset for this instance must approximately preserve the total weight placed on each basis vector. If too few copies of some e_i are stored, an adversary can zero out the largest f of them and remove too much mass at e_i. A filtering decoder cannot recreate this mass, so the decoded set fails to be a coreset. The full proof appears in Section A.2.

4 Computing Approximate Sensitivities for k-Means

The decoder requires approximate pointwise sensitivities on weighted summaries. We obtain these through a reduction to constrained weighted k-means with admissible center sets. We focus on finite metrics, where any approximation algorithm for this constrained problem yields sensitivity estimates with the same approximation factor. For Euclidean instances, one may obtain the same kind of oracle by first restricting centers to a sufficiently small candidate set, incurring only a constant factor loss.

We next fix notation. Let (X, d) be a finite metric space and let $P \subseteq X$ be the input point set. We use the weighted cost, sensitivity, and approximation conventions from Section 3. For readability, we present the argument for unweighted inputs; the weighted case follows by replacing each point contribution by its weighted contribution.

We use the following constrained version of finite-metric k-means: given a client set $P \subseteq X$ and a set $Y \subseteq X$ of admissible centers, the goal is to choose k centers $C \subseteq Y$ minimizing $\sum_{p \in P} d(p, C)^2$.

Theorem 4.1 (Finite Metrics)**.** *Suppose there is a ρ-approximation algorithm for constrained finite-metric k-means. Given a point $p \in P$, one can compute a ρ-approximation to $\sigma_P(p)$ using $O(|X|)$ calls to this algorithm.*

Proof. Fix $p \in P$, and let $S^\star$ be a set of centers maximizing the sensitivity of p. Let $x^\star \in S^\star$ be the closest center to p, and write $r^\star = d(p, x^\star)$. If we knew $r^\star$, then the best way to maximize the ratio defining the sensitivity, among solutions whose closest center to p is at distance at least $r^\star$, would be to minimize the denominator. Thus the problem reduces to a constrained k-means instance in which all centers must lie outside the ball of radius $r^\star$ around p.

Of course, we do not know $r^\star$. Since centers are chosen from the finite set X, the value of $r^\star$ is one of the distances from p to a point of X. We therefore try all such radii. For each radius r, we run the assumed ρ-approximation algorithm with admissible centers restricted to points at distance at least r from p. Let S_r be the returned center set, with cost $\widetilde{\mathrm{OPT}}_p(r) = \mathrm{cost}(P, S_r)$, and define $\widetilde{\sigma}(p)$ to be the largest value of $r^2/\widetilde{\mathrm{OPT}}_p(r)$ over all radii tried.

Every returned center set S_r satisfies $d(p, S_r) \geq r$. Hence

$$\frac{r^2}{\widetilde{\mathrm{OPT}}_p(r)} \leq \frac{d(p, S_r)^2}{\mathrm{cost}(P, S_r)} \leq \sigma_P(p),$$

so $\widetilde{\sigma}(p) \leq \sigma_P(p)$. Conversely, $S^\star$ is feasible for the constrained instance at radius $r^\star$, so the approximation guarantee gives $\widetilde{\mathrm{OPT}}_p(r^\star) \leq \rho \mathrm{cost}(P, S^\star)$. Therefore,

$$\widetilde{\sigma}(p) \geq \frac{(r^\star)^2}{\widetilde{\mathrm{OPT}}_p(r^\star)} \geq \frac{(r^\star)^2}{\rho \mathrm{cost}(P, S^\star)} = \frac{\sigma_P(p)}{\rho}.$$

Together, these inequalities show that $\widetilde{\sigma}(p)$ is a ρ-approximation to $\sigma_P(p)$.

The algorithm tries at most $|X|$ radii, and therefore makes $O(|X|)$ calls to the constrained k-means approximation algorithm. □

We complement the algorithmic reduction above with a matching hardness result. The proof is deferred to Section B.

Theorem 4.2 (Hardness of approximating k-means sensitivities)**.** *Fix $\rho > 1$. If k-Means is NP-hard to approximate within a factor ρ, then for every constant $0 < \delta < \rho - 1$, it is NP-hard to approximate the k-means sensitivity maximization problem in finite metrics within a factor $\rho - \delta$.*

Together with the hardness of k-means Guha and Khuller [1998], this implies that pointwise k-means sensitivity approximation is NP-hard within every factor $\gamma < 1 + 8/e$.

5 Experiments

In this section, we give a detailed overview of our experimental results. Codebase and results can be found here. All experiments were run locally on an Apple-silicon CPU.

Datasets. We use four standard datasets in our experiments: Adult Becker and Kohavi [1996], MNIST LeCun et al. [1998], CoverType Blackard [1998], and Taxi Moreira-Matias et al. [2013]; see table 1. In addition, we use synthetic datasets in some experiments to isolate specific failure modes of different coreset constructions.

Evaluation of Coresets. Coresets are difficult to evaluate, so we use the evaluation method initially proposed by Schwiegelshohn and Sheikh-Omar [2022]. It consists of comparing the cost they have with respect to a diverse set of solutions with the corresponding cost on the original dataset. The solution sets are constructed from k-means++-type solutions and problem-specific queries; in tractable cases, we evaluate exactly over the stated query domain. For an evaluated solution set $\mathcal{Q}$, we report the distortion $\max_{S\in\mathcal{Q}} \max\left(\frac{\text{cost}(\Omega,S)}{\text{cost}(P,S)}, \frac{\text{cost}(P,S)}{\text{cost}(\Omega,S)}\right)$, i.e. values close to 1 indicate that the coreset accurately preserves the k-means objective over the evaluated solutions.

Experiments. Broadly speaking, we consider three experiments: (i) an evaluation of the distortion of fault tolerant sensitivity sampling coresets with random and adversarial faults, contrasting Gaussian perturbations motivated by Huang et al. [2025b] with our adversarial setting, (ii) a comparison between sensitivity sampling and two commonly used coreset implementations BICO by Fichtenberger et al. [2013] and StreamKM++ by Ackermann et al. [2012] for adversarial faults, and (iii) an empirical comparison of coreset quality using exact finite-domain sensitivities and practical sensitivity estimates, motivated by Section 4, against commonly used coarser sensitivity proxies such as those implemented and evaluated in Draganov et al. [2024], Schwiegelshohn and Sheikh-Omar [2022]. We focus on the second set of experiments here; additional details as well as a full description of the first and third set of experiments are in Appendix C.

Fault-tolerance of practical coreset algorithms. We compare the fault-tolerance of sensitivity sampling with that of BICO and StreamKM++. In the worst case, BICO and StreamKM++ are not fault tolerant, but worst case bounds are notoriously loose in practice.

For all algorithms, we perform an adversarial corruption by first computing a high quality solution $\mathcal{S}$ and then computing a coreset Ω. We construct common queries by replacing one center of $\mathcal{S}$ at a time with a remote unused center, retaining exactly k centers. For each query, the adversary sets to zero the weights of the f coreset entries with largest weighted cost contributions. We report the maximum witnessed distortion over these query–corruption pairs. On 300,000 Taxi endpoints, we use $k \in \{50, 100\}$ and a coreset size cap of 2,000, matching actual entry counts across algorithms in each trial.

Across our tested adversarial corruption settings, sensitivity sampling generally achieved the lowest mean witnessed distortion. On Taxi with $k = 50$ and $f = 5$, the mean distortion over 20 trials was 1.124 for sensitivity sampling, compared with 1.689 for BICO and 1.686 for StreamKM++. The corresponding clean distortions were 1.091, 1.018, and 1.004. The advantage persisted on geographically bounded Taxi endpoints and narrowed at larger corruption budgets; see Figure 2. On MNIST, BICO already had substantial clean distortion, so its post-corruption error does not isolate the effect of faults.

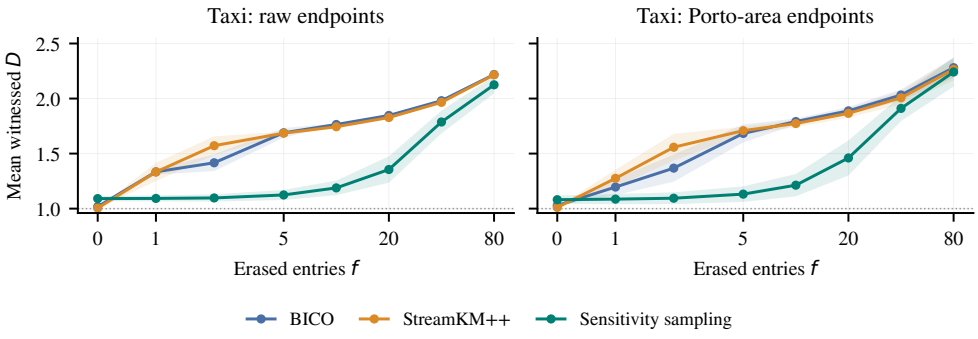


Figure 2: Comparison between BICO, StreamKM++ and sensitivity sampling on Taxi and geographically bounded Taxi endpoints, with $k = 50$. Lines show means over 20 trials; shading indicates one standard deviation.

Acknowledgments and Disclosure of Funding

Milind Prabhu was supported by NSF grant CCF-2327011. Chris Schwiegelshohn and Sudarshan Shyam were supported in part by a Google Research Award. The authors declare no additional competing financial interests.

References

Marcel R. Ackermann, Marcus Märtens, Christoph Raupach, Kamil Swierkot, Christiane Lammersen, and Christian Sohler. Streamkm++: A clustering algorithm for data streams. *ACM J. Exp. Algorithmics*, 17(1), 2012. doi: 10.1145/2133803.2184450. URL https://doi.org/10.1145/2133803.2184450.

Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Approximating extent measures of points. *J. ACM*, 51(4):606, 2004.

Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Geometric approximation via coresets. In *Combinatorial and computational geometry, MSRI*, pages 1–30. University Press, 2005.

Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. In *Proceedings of the 61st IEEE Annual Symposium on Foundations of Computer Science*, FOCS, pages 61–72, 2020.

Haris Angelidakis, Konstantin Makarychev, and Yury Makarychev. Algorithms for stable and perturbation-resilient problems. In *Symposium on Theory of Computing, STOC*, pages 438–451, 2017. doi: 10.1145/3055399.3055487. URL https://doi.org/10.1145/3055399.3055487.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, pages 1027–1035, 2007.

Pranjal Awasthi, Avrim Blum, and Or Sheffet. Stability yields a PTAS for k-median and k-means clustering. In *Symposium on Foundations of Computer Science, FOCS*, pages 309–318, 2010. doi: 10.1109/FOCS.2010.36. URL http://dx.doi.org/10.1109/FOCS.2010.36.

Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The hardness of approximation of euclidean k-means. In Lars Arge and János Pach, editors, *31st International Symposium on Computational Geometry, SoCG 2015, Eindhoven, The Netherlands, June 22-25, 2015*, LIPIcs, pages 754–767. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi: 10.4230/LIPICS.SOCG.2015.754. URL https://doi.org/10.4230/LIPIcs.SOCG.2015.754.

Olivier Bachem, Mario Lucic, and Andreas Krause. Scalable k -means clustering via lightweight coresets. In *Knowledge Discovery & Data Mining, KDD*, pages 1119–1127, 2018. doi: 10.1145/3219819.3219973. URL https://doi.org/10.1145/3219819.3219973.

Nikhil Bansal, Ola Svensson, and Luca Trevisan. New notions and constructions of sparsification for graphs and hypergraphs. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 910–928. IEEE Computer Society, 2019. doi: 10.1109/FOCS.2019.00059. URL https://doi.org/10.1109/FOCS.2019.00059.

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 *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 1707–1723. IEEE, 2024. doi: 10.1109/FOCS61266.2024.00106. URL https://doi.org/10.1109/FOCS61266.2024.00106.

Luca Becchetti, Marc Bury, Vincent Cohen-Addad, Fabrizio Grandoni, and Chris Schwiegelshohn. Oblivious dimension reduction for k-means: beyond subspaces and the johnson-lindenstrauss lemma. In *Symposium on Theory of Computing, STOC*, pages 1039–1050, 2019. URL https://doi.org/10.1145/3313276.3316318.

Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. URL https://doi.org/10.24432/C5XW20.

Jock Blackard. Covertype. UCI Machine Learning Repository, 1998. URL https://doi.org/10.24432/C50K5N.

Guy Blanc and Gregory Valiant. Adaptive and oblivious statistical adversaries are equivalent. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC. ACM, 2025. doi: 10.1145/3717823.3718193. Also available as arXiv:2410.13548.

Guy Blanc, Jane Lange, Ali Malik, and Li-Yang Tan. On the power of adaptivity in statistical adversaries. In *Proceedings of the 35th Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 5030–5061. PMLR, 2022.

Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal coresets for least-squares regression. *IEEE Trans. Inf. Theory*, 59(10):6880–6892, 2013. doi: 10.1109/TIT.2013.2272457. URL https://doi.org/10.1109/TIT.2013.2272457.

Vladimir Braverman, Dan Feldman, and Harry Lang. New frameworks for offline and streaming coreset constructions. *CoRR*, abs/1612.00889, 2016. URL http://arxiv.org/abs/1612.00889.

Vladimir Braverman, Shaofeng H-C Jiang, Robert Krauthgamer, and Xuan Wu. Coresets for clustering in excluded-minor graphs and beyond. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2679–2696. SIAM, 2021.

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 *Symposium on Foundations of Computer Science, FOCS*, pages 462–473, 2022. doi: 10.1109/FOCS54457.2022.00051. URL https://doi.org/10.1109/FOCS54457.2022.00051.

Jarosław Byrka, Yuhao Guo, Yang Hu, Shi Li, Chengzhang Wan, and Zaixuan Wang. k-clustering via iterative randomized rounding. *arXiv preprint arXiv:2604.06046*, 2026.

Clément L. Canonne, Samuel B. Hopkins, Jerry Li, Allen Liu, and Shyam Narayanan. The full landscape of robust mean testing: Sharp separations between oblivious and adaptive contamination. In *Proceedings of the 64th IEEE Annual Symposium on Foundations of Computer Science*, FOCS, pages 36–45. IEEE, 2023.

Moses Charikar, Vincent Cohen-Addad, Ruiquan Gao, Fabrizio Grandoni, Euiwoong Lee, and Ernest van Wijland. An improved greedy approximation for (metric) k-means. In *66th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2025, Sydney, Australia, December 14-17, 2025*, pages 233–240. IEEE, 2025. doi: 10.1109/FOCS63196.2025.00016. URL https://doi.org/10.1109/FOCS63196.2025.00016.

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.

Michael B. Cohen, Cameron Musco, and Christopher Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1758–1777, 2017.

Vincent Cohen-Addad and Karthik C. S. Inapproximability of clustering in lp metrics. In *Symposium on Foundations of Computer Science, FOCS*, pages 519–539, 2019. doi: 10.1109/FOCS.2019.00040. URL https://doi.org/10.1109/FOCS.2019.00040.

Vincent Cohen-Addad and Chris Schwiegelshohn. On the local structure of stable clustering instances. In *Symposium on Foundations of Computer Science, FOCS*, pages 49–60, 2017. doi: 10.1109/FOCS.2017.14. URL https://doi.org/10.1109/FOCS.2017.14.

Vincent Cohen-Addad, Philip N. Klein, and Claire Mathieu. The power of Local Search for clustering. *CoRR*, abs/1603.09535, 2016. URL http://arxiv.org/abs/1603.09535.

Vincent Cohen-Addad, Andreas Emil Feldmann, and David Saulpic. Near-linear time approximation schemes for clustering in doubling metrics. *J. ACM*, 68(6):44:1–44:34, 2021a. doi: 10.1145/3477541. URL https://doi.org/10.1145/3477541.

Vincent Cohen-Addad, Karthik C. S., and Euiwoong Lee. On approximability of clustering problems without candidate centers. In *Symposium on Discrete Algorithms, SODA*, pages 2635–2648, 2021b. doi: 10.1137/1.9781611976465.156. URL https://doi.org/10.1137/1.9781611976465.156.

Vincent Cohen-Addad, David Saulpic, and Chris Schwiegelshohn. A new coreset framework for clustering. In *Symposium on Theory of Computing, STOC*, pages 169–182, 2021c. doi: 10.1145/3406325.3451022. URL https://doi.org/10.1145/3406325.3451022.

Vincent Cohen-Addad, Hossein Esfandiari, Vahab S. Mirrokni, and Shyam Narayanan. Improved approximations for euclidean k-means and k-median, via nested quasi-independent sets. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1621–1628. ACM, 2022a. doi: 10.1145/3519935.3520011. URL https://doi.org/10.1145/3519935.3520011.

Vincent Cohen-Addad, Karthik C. S., and Euiwoong Lee. Johnson coverage hypothesis: Inapproximability of k-means and k-median in ℓ_p-metrics. In *Symposium on Discrete Algorithms, SODA*, pages 1493–1530, 2022b. doi: 10.1137/1.9781611977073.63. URL https://doi.org/10.1137/1.9781611977073.63.

Vincent Cohen-Addad, Kasper Green Larsen, David Saulpic, and Chris Schwiegelshohn. Towards optimal lower bounds for k-median and k-means coresets. In *Symposium on Theory of Computing, STOC*, pages 1038–1051, 2022c. doi: 10.1145/3519935.3519946. URL https://doi.org/10.1145/3519935.3519946.

Vincent Cohen-Addad, Kasper Green Larsen, David Saulpic, Chris Schwiegelshohn, and Omar Ali Sheikh-Omar. Improved coresets for euclidean k-means. In *NeurIPS*, 2022d.

Vincent Cohen-Addad, David Saulpic, and Chris Schwiegelshohn. Deterministic clustering in high dimensional spaces: Sketches and approximation. *CoRR*, abs/2310.04076, 2023. doi: 10.48550/ARXIV.2310.04076. URL https://doi.org/10.48550/arXiv.2310.04076.

Vincent Cohen-Addad, Andrew Draganov, Matteo Russo, David Saulpic, and Chris Schwiegelshohn. A tight vc-dimension analysis of clustering coresets with applications. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 4783–4808. SIAM, 2025. doi: 10.1137/1.9781611978322.162. URL https://doi.org/10.1137/1.9781611978322.162.

Vincent Cohen-Addad, Karthik C. S., David Saulpic, and Chris Schwiegelshohn. Near-optimal bounds for parameterized euclidean k-means. *CoRR*, abs/2603.28268, 2026. doi: 10.48550/ARXIV.2603.28268. URL https://doi.org/10.48550/arXiv.2603.28268.

Anirban Dasgupta, Petros Drineas, Boulos Harb, Ravi Kumar, and Michael W. Mahoney. Sampling algorithms and coresets for ℓ_p regression. *SIAM J. Comput.*, 38(5):2060–2078, 2009.

Ilias Diakonikolas and Daniel M Kane. *Algorithmic high-dimensional robust statistics*. Cambridge university press, 2023.

Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, FOCS, pages 655–664. IEEE Computer Society, 2016.

Andrew Draganov, David Saulpic, and Chris Schwiegelshohn. Settling time vs. accuracy tradeoffs for clustering big data. *Proc. ACM Manag. Data*, 2(3):173, 2024. doi: 10.1145/3654976. URL https://doi.org/10.1145/3654976.

Dan Feldman. Core-sets: An updated survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 10 (1), 2020. doi: 10.1002/widm.1335. URL https://doi.org/10.1002/widm.1335.

Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Symposium on Theory of Computing, STOC*, pages 569–578, 2011.

Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for *k*-means, PCA and projective clustering. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1434–1453, 2013.

Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering. *SIAM J. Comput.*, 49(3):601–657, 2020. doi: 10.1137/18M1209854. URL https://doi.org/10.1137/18M1209854.

Zhili Feng, Praneeth Kacham, and David P. Woodruff. Dimensionality reduction for the sum-of-distances metric. In *International Conference on Machine Learning, ICML*, pages 3220–3229, 2021. URL http://proceedings.mlr.press/v139/feng21a.html.

Hendrik Fichtenberger, Marc Gillé, Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. BICO: BIRCH meets coresets for k-means clustering. In Hans L. Bodlaender and Giuseppe F. Italiano, editors, *Algorithms - ESA 2013 - 21st Annual European Symposium, Sophia Antipolis, France, September 2-4, 2013. Proceedings*, 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.

Zachary Friggstad, Mohsen Rezapour, and Mohammad R. Salavatipour. Local search yields a PTAS for k-means in doubling metrics. *CoRR*, abs/1603.08976, 2016. URL http://arxiv.org/abs/1603.08976.

Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, pages 649–657, 1998.

Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Symposium on Theory of Computing, STOC*, pages 291–300, 2004.

Lingxiao Huang and Nisheeth K. Vishnoi. Coresets for clustering in euclidean spaces: importance sampling is nearly optimal. In *Symposium on Theory of Computing, STOC 2020*, 2020. doi: 10.1145/3357713.3384296. URL https://doi.org/10.1145/3357713.3384296.

Lingxiao Huang, Shaofeng H.-C. Jiang, Jian Li, and Xuan Wu. ε-coresets for clustering in doubling metrics. In *Proceedings of the 59th IEEE Annual Symposium on Foundations of Computer Science*, FOCS, pages 814–825, 2018.

Lingxiao Huang, Shaofeng H.-C. Jiang, Jianing Lou, and Xuan Wu. Near-optimal coresets for robust clustering. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=Nc1ZkRW8Vde.

Lingxiao Huang, Jian Li, and Xuan Wu. On optimal coreset construction for euclidean (k, z)-clustering, 2024. URL https://doi.org/10.1145/3618260.3649707.

Lingxiao Huang, Zhenyu Jiang, Yi Li, and Xuan Wu. On tight robust coresets for *k*-medians clustering. *CoRR*, abs/2507.11260, 2025a. doi: 10.48550/ARXIV.2507.11260. URL https://doi.org/10.48550/arXiv.2507.11260.

Lingxiao Huang, Zhize Li, Nisheeth K. Vishnoi, Runkai Yang, and Haoyu Zhao. Coresets for clustering under stochastic noise. *CoRR*, abs/2510.23438, 2025b. doi: 10.48550/ARXIV.2510.23438. URL https://doi.org/10.48550/arXiv.2510.23438.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. doi: 10.1214/aoms/1177703732.

Chansophea Wathanak In, Yi Li, David Woodruff, and Xuan Wu. Robust sparsification via sensitivity. In *International Conference on Machine Learning*, pages 26446–26463. PMLR, 2025.

Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM*, 48(2):274–296, 2001.

Arun Jambulapati, Yang P. Liu, and Aaron Sidford. Chaining, group leverage score overestimates, and fast spectral hypergraph sparsification. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 196–206. ACM, 2023. doi: 10.1145/3564246.3585136. URL https://doi.org/10.1145/3564246.3585136.

Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. *Computational Geometry*, 28(2–3):89–112, 2004.

Kevin A. Lai, Anup B. Rao, and Santosh S. Vempala. Agnostic estimation of mean and covariance. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, FOCS, pages 665–674. IEEE Computer Society, 2016. doi: 10.1109/FOCS.2016.76.

Michael Langberg and Leonard J. Schulman. Universal ϵ-approximators for integrals. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, pages 598–607, 2010.

Yann LeCun, Corinna Cortes, and Christopher J. C. Burges. MNIST Database of Handwritten Digits. UCI Machine Learning Repository, 1998. URL https://doi.org/10.24432/C53K8Q.

Euiwoong Lee, Melanie Schmidt, and John Wright. Improved and simplified inapproximability for k-means. *Inf. Process. Lett.*, 120:40–43, 2017. doi: 10.1016/J.IPL.2016.11.009. URL https://doi.org/10.1016/j.ipl.2016.11.009.

James R. Lee. Spectral hypergraph sparsification via chaining. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 207–218. ACM, 2023. doi: 10.1145/3564246.3585165. URL https://doi.org/10.1145/3564246.3585165.

Yin Tat Lee and He Sun. Constructing linear-sized spectral sparsification in almost-linear time. *SIAM J. Comput.*, 47(6):2315–2336, 2018. doi: 10.1137/16M1061850. URL https://doi.org/10.1137/16M1061850.

Gábor Lugosi and Shahar Mendelson. Mean estimation and regression under heavy-tailed distributions: A survey. *Found. Comput. Math.*, 19(5):1145–1190, 2019. doi: 10.1007/s10208-019-09427-x. URL https://doi.org/10.1007/s10208-019-09427-x.

Alaa Maalouf, Adiel Statman, and Dan Feldman. Tight sensitivity bounds for smaller coresets. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2051–2061, 2020.

Tung Mai, Cameron Musco, and Anup Rao. Coresets for classification - simplified and strengthened. In *Neural Information Processing Systems (NeurIPS)*, pages 11643–11654, 2021.

Shannon R. McCurdy. Ridge regression and provable deterministic ridge leverage score sampling. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2468–2477, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/e1d5be1c7f2f456670de3d53c7b54f4a-Abstract.html.

Luis Moreira-Matias, Michel Ferreira, Joao Mendes-Moreira, L. L., and J. J. Taxi Service Trajectory – Prediction Challenge, ECML PKDD 2015. UCI Machine Learning Repository, 2013. URL https://doi.org/10.24432/C55W25.

Alexander Munteanu and Simon Omlor. Optimal bounds for ℓ_p sensitivity sampling via ℓ_2 augmentation. In *Proceedings of the 41st International Conference on Machine Learning*, pages 36769–36796. PMLR, 2024. URL https://proceedings.mlr.press/v235/munteanu24a.html.

Alexander Munteanu, Chris Schwiegelshohn, Christian Sohler, and David P. Woodruff. On coresets for logistic regression. In *Neural Information Processing Systems, NeurIPS*, pages 6562–6571, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/63bfd6e8f26d1d3537f4c5038264ef36-Abstract.html.

R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The effectiveness of Lloyd-type methods for the k-means problem. *J. ACM*, 59(6):28, 2012. doi: 10.1145/2395116.2395117. URL http://doi.acm.org/10.1145/2395116.2395117.

Dimitris S. Papailiopoulos, Anastasios Kyrillidis, and Christos Boutsidis. Provable deterministic leverage score sampling. In Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani, editors, *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 997–1006. ACM, 2014. doi: 10.1145/2623330.2623698. URL https://doi.org/10.1145/2623330.2623698.

Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. Fair coresets and streaming algorithms for fair k-means. In *Approximation and Online Algorithms, WAOA*, pages 232–251, 2019. doi: 10.1007/978-3-030-39479-0_16. URL https://doi.org/10.1007/978-3-030-39479-0_16.

Chris Schwiegelshohn and Omar Ali Sheikh-Omar. An empirical evaluation of k-means coresets. In *European Symposium on Algorithms, ESA*, volume 244 of *LIPIcs*, pages 84:1–84:17, 2022. doi: 10.4230/LIPIcs.ESA.2022.84. URL https://doi.org/10.4230/LIPIcs.ESA.2022.84.

Christian Sohler and David P. Woodruff. Strong coresets for k-median and subspace approximation: Goodbye dimension. *CoRR*, abs/1809.02961, 2018. URL http://arxiv.org/abs/1809.02961.

Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM J. Comput.*, 40(6):1913–1926, 2011. doi: 10.1137/080734029. URL https://doi.org/10.1137/080734029.

John W. Tukey. A survey of sampling from contaminated distributions. In Ingram Olkin, Sudhist G. Ghurye, Wassily Hoeffding, William G. Madow, and Henry B. Mann, editors, *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, pages 448–485. Stanford University Press, 1960.

Kasturi R. Varadarajan and Xin Xiao. On the sensitivity of shape fitting problems. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, Hyderabad, India, December 15-17, 2012*, LIPIcs, pages 486–497. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012. doi: 10.4230/LIPICS.FSTTCS.2012.486. URL https://doi.org/10.4230/LIPIcs.FSTTCS.2012.486.

David P. Woodruff and Taisuke Yasuda. Coresets for multiple ℓ_p regression. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 53202–53233. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/woodruff24a.html.

A Omitted Proofs from Section 3

A.1 Proxy Sensitivity Upper Bounds

Typically, k-means coreset construction algorithms do not sample points proportional to exact sensitivities. Instead, points are sampled using proxy scores which are upper bounds on the sensitivity scores Feldman and Langberg [2011], Feldman et al. [2020], Huang and Vishnoi [2020], Draganov et al. [2024], Bansal et al. [2024]. Their appeal is twofold: they often lead to a cleaner analysis, and the proxy scores are more easily computable.

For example a typical proxy score one might use is as follows. One can first compute a constant-factor approximate solution $A = \{a_1, \ldots, a_k\}$ and let C_j be the cluster assigned to a_j. For each $p \in P$,

$$\tau_P(p) = \Theta\left(\frac{1}{|C_j|} + \frac{\mathrm{cost}(p, A)}{\mathrm{cost}(P, A)}\right) \qquad \text{for } p \in C_j.$$

These scores are always upper bounds for the true sensitivities. However, they might be significantly overestimated. For example, in a high dimensional simplex, a good clustering may contain single points, and these will have their sensitivity overestimated by as much as a factor n.

We now show that our framework also applies to coreset constructions in which points are sampled proportional to such proxy scores.

Let $\tau : P \to \mathbb{R}_{\geq 0}$ satisfy $\tau(p) \geq \sigma_P(p)$ for every $p \in P$, and let $T_\tau := \sum_{p \in P} \tau(p)$. Suppose the initial coreset is constructed by sampling each point p with probability $q_p = \tau(p)/T_\tau$ and assigning each sampled copy weight $1/(mq_p)$.

Theorem A.1 (Fault tolerance from proxy sensitivities)**.** *Fix $\varepsilon \in (0, 1/10)$ and $f \in \mathbb{N}$. Let $m_{\mathrm{sc}}(\varepsilon)$ be the sample size required by the above sampling scheme to obtain an ε-coreset for P with constant probability. Assume that α-approximate sensitivities can be computed efficiently for any point in any weighted set. If Ω is constructed with*

$$m = m_{\mathrm{sc}}(\varepsilon) + \frac{\alpha f T_\tau}{\varepsilon}$$

samples, then the filtering decoder of Algorithm 1 outputs a 6ε-coreset for every f-fault corruption $\widetilde{\Omega}$ of Ω, with constant probability over the construction of Ω.

Proof. The proof is identical to the proof of Theorem 3.1, with Lemma 3.2 replaced by the following bound. For any sampled copy x, its weight is $w(x) = T_\tau/(m\tau(x))$. Since $\tau(x) \geq \sigma_P(x)$, for every solution S,

$$w(x)\ell_x(S) = \frac{T_\tau}{m\tau(x)}\ell_x(S) \leq \frac{T_\tau}{m\tau(x)}\sigma_P(x)L_P(S) \leq \frac{T_\tau}{m}L_P(S).$$

Thus every occurrence of the bound $TL_P(S)/m$ in the proof of Theorem 3.1 is replaced by $T_\tau L_P(S)/m$. The rest of the peeling analysis is unchanged, giving the claimed guarantee. □

A natural question is whether these proxy upper bounds can also replace the pointwise sensitivity approximations in the decoder. The following example shows that a constant-factor approximation guarantee for the reference clustering does not suffice: its proxy scores can strictly favor deleting a genuine point over a harmful corruption.

Proxy scores are not enough for decoding. Fix $\varepsilon \in (0, 1/10)$ and an integer $n \geq 8/\varepsilon$. Consider 2-means in the shortest-path metric of the following graph. The input $P = \{p_1, \ldots, p_n\}$ forms a complete graph with unit edge lengths. Attach a vertex a to p_1 by an edge of length $1/2$, and join a vertex b to every p_i by an edge of length M, where $M^2 = n - 4$. Centers may be any vertices. Take the exact clean summary $\Omega = P$, with unit weights. Every two-center query assigns the same squared cost to all but at most two input points, whose costs can only be smaller. Hence each input point has sensitivity $1/(n-2)$, so the clean summary has uniformly small pointwise contributions.

Corrupt p_n by moving it to b, and call the resulting summary Q. The reference centers $A = \{a, p_2\}$ induce a singleton cluster $\{p_1\}$ and a cluster containing all other points, including b. Their cost is

$D := \text{cost}(Q, A) = n - 3 + 1/4 + M^2 = 2n - 27/4$. Any two-center solution costs at least $n - 3$ on the $n - 1$ genuine points, so A is a 2-approximation.

Consider the usual proxy scores, scaled by a common factor:

$$\tau_Q(x) = 2\left(\frac{1}{|C(x)|} + \frac{d(x, A)^2}{D}\right).$$

The genuine singleton has score $\tau_Q(p_1) = 2 + 1/(2D) > 2$, whereas

$$1 < \tau_Q(b) = \frac{2}{n-1} + \frac{2(n-4)}{D} < 2,$$

and every other score is also less than 2. These scores are valid sensitivity upper bounds: every genuine entry q satisfies $\sigma_Q(q) \le 1/(n-3) \le 2/(n-1) \le \tau_Q(q)$, by the same common-cost observation, and $\sigma_Q(b) \le 1 < \tau_Q(b)$.

Thus the proxy-based decoder uniquely deletes p_1 and leaves the corruption. For $S = \{p_2, p_3\}$, the decoded summary $\Omega^\star$ has distortion

$$\frac{\text{cost}(\Omega^\star, S)}{\text{cost}(P, S)} = \frac{2(n-4)}{n-2} \ge 2 - \varepsilon > 1 + 6\varepsilon.$$

In contrast, the same query shows that $\sigma_Q(b) \ge (n-4)/(2n-7) > 1/(n-3)$, so the exact-sensitivity decoder uniquely deletes b. Its output is $P \setminus \{p_n\}$, an ε-coreset because removing one point loses at most a $1/(n-2) \le \varepsilon$ fraction of any query cost. The proxy rule therefore fails even on an exact clean summary whose individual contributions are small.

A.2 Lower Bound for Fault-Tolerant k-Means Coresets

In this section, we prove Theorem 3.4. The construction is the simple k-means instance $P = \{e_1, \ldots, e_k\} \subset \mathbb{R}^k$. The proof shows that a filtering decoder cannot recover from f corruptions unless the coreset stores enough separate mass at each coordinate point.

For a weighted multiset C supported on P, write

$$W_i(C) := \sum_{q \in C:\, q = e_i} w(q)$$

for the total weight assigned to copies of e_i.

Claim A.2 (Weight preservation)**.** *If C is an ε-coreset for P, then $W_i(C) \in [1 - \varepsilon, 1 + \varepsilon]$ for all $i \in [k]$.*

Proof. Fix $i \in [k]$ and take

$$S_i := \{e_1, \ldots, e_{i-1}, e_{i+1}, \ldots, e_k, 0\}.$$

Then $\text{cost}(P, S_i) = \|e_i - 0\|^2 = 1$. Moreover, $\text{cost}(C, S_i) = W_i(C)$, since only copies of e_i contribute to the cost, each with squared distance 1. Since C is an ε-coreset for P,

$$W_i(C) = \text{cost}(C, S_i) \in (1 \pm \varepsilon)\text{cost}(P, S_i) = [1 - \varepsilon, 1 + \varepsilon].$$

□

Proof of Theorem 3.4. Assume for contradiction that $|\Omega| < fk/(2\varepsilon)$. By pigeonhole, there is some $i \in [k]$ such that Ω contains fewer than $f/(2\varepsilon)$ copies of e_i. Let their weights be $a_1, \ldots, a_m$, where $m < f/(2\varepsilon)$, and let

$$A := \sum_{j=1}^{m} a_j = W_i(\Omega).$$

Since Ω is an ε-coreset, Claim A.2 gives $A \in [1 - \varepsilon, 1 + \varepsilon]$.

Let J be the set of the $\min\{f, m\}$ largest weights among $a_1, \ldots, a_m$. We claim that

$$\sum_{j \in J} a_j > 2\varepsilon A.$$

Indeed, if $m \le f$, then $J = [m]$ and the sum is $A > 2\varepsilon A$ because $\varepsilon < 1/2$. If $m > f$, then the average weight among the m copies is A/m, so the sum of the f largest weights is at least

$$f \cdot \frac{A}{m} > f \cdot \frac{A}{f/(2\varepsilon)} = 2\varepsilon A.$$

The adversary corrupts the entries indexed by J by setting their weights to 0, obtaining an f-fault corruption $\widetilde{\Omega}$. Let $\Omega^\star := \textsc{Dec}(\widetilde{\Omega})$. Since the decoder is a filtering decoder, $\Omega^\star \subseteq \widetilde{\Omega}$ and inherits weights. Therefore,

$$W_i(\Omega^\star) \le W_i(\widetilde{\Omega}) = A - \sum_{j \in J} a_j < (1 - 2\varepsilon)A \le (1 - 2\varepsilon)(1 + \varepsilon) < 1 - \varepsilon.$$

By Claim A.2, $\Omega^\star$ cannot be an ε-coreset for P. This contradicts the definition of an f-fault-tolerant ε-coreset. □

B Approximating Sensitivities: Omitted Proofs from Section 4

In this section, we prove Theorem 4.2, as well as give an overview of approximation algorithms and hardness of approximation and the implications on computing sensitivities. The reduction is from the gap version of finite-metric k-means.

Definition B.1 (ρ-Gap k-Means Problem)**.** Given a finite-metric k-means instance and a threshold $B > 0$, decide whether the optimum k-means cost is at most B or greater than ρB, assuming one of these is the case.

It is known that, in general metrics, k-means is NP-hard to approximate within a factor $1 + 8/e - \eta$ for every constant $\eta > 0$ Guha and Khuller [1998]. Equivalently, for every constant $\rho < 1 + 8/e$, the ρ-Gap k-Means problem is NP-hard.

Proof of Theorem 4.2. Let $\alpha := \rho - \delta$. Consider a finite-metric k-means instance given by a point set P in a metric space (X, d). Let OPT denote its optimum k-means cost, and let D denote the diameter of X. We may assume $D > 0$.

Construct a new metric space $X' = X \cup \{p^\star\}$ by adding a point $p^\star$ with $d(p^\star, x) = D$ for every $x \in X$. Since D is the diameter of X, this defines a metric on X'. Assign weight L to every point of P, where L will be chosen later, and add $p^\star$ with unit weight. Let P' be the resulting weighted instance.

We first relate the sensitivity of $p^\star$ in P' to the optimum value of the original k-means instance. We claim that

$$\sigma^*(p^\star) = \frac{D^2}{D^2 + L\text{OPT}}.$$

Indeed, let S be any set of k centers in X'. If $p^\star \in S$, then $\text{cost}(p^\star, S) = 0$, so the corresponding ratio is zero. Otherwise all centers lie in X, and therefore $\text{cost}(p^\star, S) = D^2$ and

$$\text{cost}(P', S) = D^2 + L\text{cost}(P, S).$$

Thus

$$\frac{\text{cost}(p^\star, S)}{\text{cost}(P', S)} = \frac{D^2}{D^2 + L\text{cost}(P, S)}.$$

This ratio is maximized by minimizing $\text{cost}(P, S)$, proving the claim.

Now suppose there is an α-approximation algorithm for sensitivity maximization. Define

$$A := \frac{D^2}{D^2 + LB}.$$

If OPT $\le B$, then $\sigma^*(p^\star) \ge A$, so the algorithm returns a solution of value at least A/α.

On the other hand, if OPT $> \rho B$, then

$$\sigma^*(p^\star) < \frac{D^2}{D^2 + L\rho B}.$$

Choose L large enough so that

$$\alpha(D^2 + LB) < D^2 + L\rho B.$$

This is possible because $\alpha = \rho - \delta < \rho$. Hence

$$\sigma^*(p^\star) < \frac{D^2}{D^2 + L\rho B} < \frac{D^2}{\alpha(D^2 + LB)} = \frac{A}{\alpha}.$$

Thus, by thresholding the algorithm's output at A/α, we distinguish the YES and NO cases of ρ-Gap k-Means. This contradicts the assumed NP-hardness of the gap problem. □

Related Work on Approximation Algorithms As mentioned in the main body, the k-means objective cannot be approximated beyond a factor $1 + 8/e$ Guha and Khuller [1998]. The problem remains hard to approximate in ℓ_p metrics including but not limited to Euclidean space, see Awasthi et al. [2015], Cohen-Addad and Karthik C. S. [2019], Cohen-Addad et al. [2022b, 2021b], Lee et al. [2017]. The approximation factor typically depends on the specific metric and, in some cases, on hardness conjectures stronger than P!=NP. Approximation algorithms for clustering in general metric spaces are an intensely studied topic Ahmadian et al. [2020], Arthur and Vassilvitskii [2007], Byrka et al. [2026], Cohen-Addad et al. [2022a], Jain and Vazirani [2001], Kanungo et al. [2004], with the current state of the art being a 5.83 approximation Charikar et al. [2025]. Further improvements are possible in constrained metrics such as low dimensional Euclidean spaces Cohen-Addad et al. [2026], planar graph metrics Cohen-Addad et al. [2016], and doubling metrics Cohen-Addad et al. [2021a], Friggstad et al. [2016], or assumptions on the input Angelidakis et al. [2017], Awasthi et al. [2010], Cohen-Addad and Schwiegelshohn [2017], Ostrovsky et al. [2012].

C Further Experimental Evaluation

Here are the datasets used in the experiments.

Dataset	Shape after preprocessing
Adult	$(48842,\ 108)$
MNIST	$(70000,\ 784)$
CoverType	$(581012,\ 54)$
Taxi	$(1704769,\ 2)$

Table 1: Datasets used with their shapes after preprocessing. Experimental subset sizes are specified below. Adult has 14 original features before encoding.

Preprocessing. For Adult, we exclude the income label, standardize the six numeric features, and one-hot encode the eight categorical features, retaining missing categories. The census sampling weight is treated as an ordinary numeric feature. For CoverType, we exclude the label and standardize all 54 features. For MNIST, we pool the training and test images, exclude labels, and divide pixel values by 255, without PCA or per-pixel standardization. For Taxi, we convert each nonempty trajectory into a two-dimensional point by taking the final GPS coordinate of the trip. The public download yields $1{,}704{,}769$ endpoints after excluding $5{,}901$ empty trajectories. We also use a geographically bounded control with longitude in $[-8.8, -8.2]$ and latitude in $[41.0, 41.4]$, containing $1{,}701{,}157$ endpoints. Taxi subsets are translated and divided by a common scalar root-mean-square radius; this preserves clean and weight-erasure cost ratios. Full-feature experiments use ambient domain $\mathbb{R}^d$, and perturbation sizes below refer to the resulting coordinates.

All input subsets are sampled without replacement. Sampling coresets draw with replacement, assigning each sampled copy of point i weight $1/(mp_i)$, where p_i is its sampling probability. Copies remain separate entries. Decoder outputs retain inherited weights, without renormalization. Unless otherwise stated, results are means over the specified confirmation trials and error bars indicate one sample standard deviation; pilot runs are kept separately. Accuracy events use a numerical tolerance of 10^{-12}.

Computational resources. All reported experiments were run locally on an Apple M1 CPU (8 cores) with 16 GB of unified memory, running macOS 26.6.2 and Python 3.12.14. Numerical libraries were limited to two threads per job; no GPU was used. Approximate summed elapsed job times were 10 minutes for fault recovery and its controls, 33 minutes for baseline comparisons, and 35 minutes for sampling-distribution comparisons, including the simplex controls, totalling approximately 1.3 job-hours. The baseline total includes the interrupted-and-resumed fixed-threshold control. Data preparation, environment setup, additional pilot and validation runs, and postprocessing are excluded; the full research required additional compute.

C.1 Fault-Tolerant Coresets

Sensitivity Sampling against random and adversarial faults. We first evaluate fault-tolerant coresets for k-means on the Adult, MNIST, CoverType and Taxi datasets. We construct a sensitivity-sampling coreset of size m from the clean data and then corrupt a small number $f \ll m$ of its entries. We consider two corruption models: random corruption, where randomly selected coreset points receive Gaussian perturbations, and adversarial corruption, where selected points are moved far away in different directions and assigned much larger weights. Motivated by the stochastic-noise setting of Huang et al. [2025b], we test Gaussian perturbations here after coreset construction, with our goal of recovering a good coreset.

For each coreset size m, we compare the clean coreset, the corrupted coreset, the fault-tolerant coreset, and a random-deletion control. The decoder repeatedly recomputes weighted sensitivity estimates on the current corrupted coreset and removes the most sensitive point for f iterations. For the full-feature $k = 10$ experiments, these estimates are exact only over a finite family generated from the corrupted summary: 128 randomly selected ten-point center sets and eight weighted or unweighted k-means solutions. This is a practical heuristic, without a certified approximation factor for unrestricted sensitivities. The decoder receives only the corrupted summary and f; clean data and corruption labels are used only for evaluation.

For downstream clustering, we run weighted k-means on each resulting coreset. All resulting center sets are evaluated on the original clean datasets, and we report the clean-data cost ratio relative to a full-data k-means reference solution. This reference is a high-quality local solution. For coreset quality evaluation, we use the evaluation approach of Schwiegelshohn and Sheikh-Omar [2022] with a separate family of 32 clean-data queries: local k-means solutions, k-means++ seeds, random input-point center sets, and perturbed seeds. Coreset distortion and downstream clustering cost ratio are reported separately.

Random corruptions. Let $\Omega = \{(x_i, w_i)\}_{i=1}^m$ be the coreset and let $\widetilde{\Omega} = \{(\widetilde{x}_i, \widetilde{w}_i)\}_{i=1}^m$ denote the corrupted coreset. We choose a corruption parameter $f \le m$ and sample a subset uniformly at random:

$$I \subseteq [m], \qquad |I| = f,$$

For every $i \notin I$, we keep the point unchanged: $\widetilde{x}_i = x_i$. For every $i \in I$, we replace it by a randomly perturbed version. After the preprocessing above, all four datasets use Gaussian perturbations:

$$\widetilde{x}_i = x_i + \sigma\xi_i, \qquad \xi_i \sim \mathcal{N}(0, I_d), \qquad \sigma = 0.2.$$

All weights are unchanged. For MNIST, we use the same additive Gaussian pixel noise, without clipping. This model tests corruptions that need not dominate the clustering objective. In high dimension, the norm of a perturbation can nevertheless be appreciable.

Adversarial corruptions. Let Ω and $\widetilde{\Omega}$ denote the clean and corrupted coresets above. We choose a corruption parameter $f \le m$ and select the f entries with largest original weights, using a fixed rule to break ties. For every $i \notin I$, the point and weight remain unchanged. For every $i \in I$, we set

$$\widetilde{x}_i = x_i + Ru_i, \qquad \widetilde{w}_i = \lambda w_i,$$

where $R = 80$, $\lambda = 10$, and the u_i are independent Gaussian directions normalized to unit length. This is an explicit admissible adversarial attack family, rather than an optimization over all possible faults. It checks whether the algorithm identifies and removes harmful points rather than allowing them to dominate the clustering objective.

Experimental Setup. The main parameters are $k = 10$ clusters, coreset sizes

$$m \in \{100, 200, 500, 1000\},$$

and corruption level $f = 0.05m$, giving $f \in \{5, 10, 25, 50\}$ corrupted coreset entries. Each setting is averaged over 20 construction trials on a fixed 5,000-point subset per dataset. The initial ten-center coresets use the standard sensitivity proxy described below. The random-deletion control removes f uniformly chosen entries from the same corrupted coreset. We plot coreset distortion as a function of m and separately report the clean-data clustering cost ratios.

Empirical Observations and Trends. The decoding algorithm recovers low empirical distortion under the specified adversarial corruptions. At $m = 1000$ and $f = 50$, mean decoded distortions are 1.088, 1.160, 1.076, and 1.161 on Adult, CoverType, MNIST, and Taxi, respectively, compared with corrupted distortions of approximately 1035, 222.3, 109.1, and 66559. The decoder removes all designated faults in these settings. The corresponding mean downstream cost ratios are 1.027, 1.037, 1.029, and 1.116.

Under random noise, decoding is often unnecessary and forced deletion can worsen coreset quality. At $m = 1000$, the noisy distortions on CoverType and Taxi are 1.056 and 1.084, while decoding increases them to 1.530 and 1.573. Their decoded downstream cost ratios remain 1.018 and 1.040. The decoder makes all f deletions even when the faults have little influence, and the discarded weight is not redistributed. Thus good downstream clustering does not imply that deletion preserves all evaluated query costs. Figures 3–10 show both regimes. Hence, our experiments show that the framework is effective for the specified adversarial corruptions, while mild noise may not require decoding.

Exact recovery checks and additional controls. To distinguish the heuristic from exact decoding, we also run $k = 1$ on the same four datasets, with 30 trials per setting. Weighted sensitivities have the closed form

$$\sigma_Q(q_i) = \frac{w_i}{W} + \frac{w_i\|q_i - \mu_Q\|^2}{V_Q}, \qquad W = \sum_i w_i, \quad V_Q = \sum_i w_i\|q_i - \mu_Q\|^2,$$

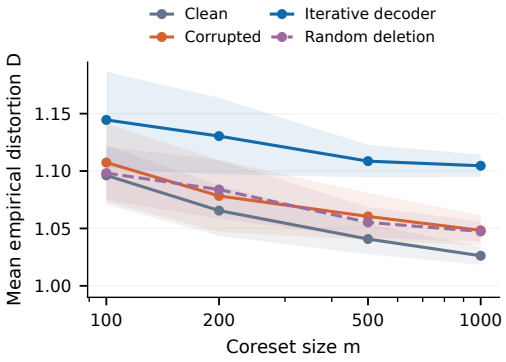


Figure 3: Distortion on Adult under random Gaussian corruptions. Means and one-standard-deviation bands over 20 trials; $k = 10$, $f = 0.05m$.

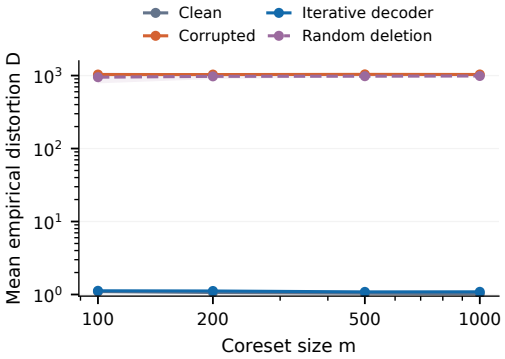


Figure 4: Distortion on Adult under adversarial coordinate and weight corruptions. Same settings as Figure 3; logarithmic vertical axis.

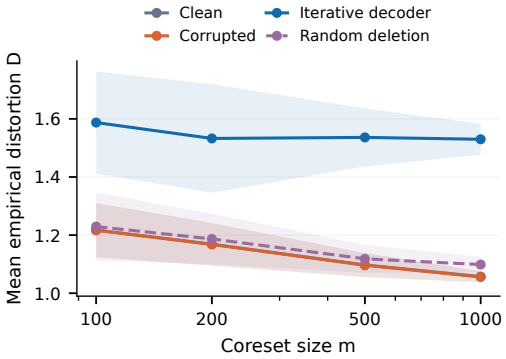


Figure 5: Distortion on CoverType under random Gaussian corruptions. Same settings as Figure 3.

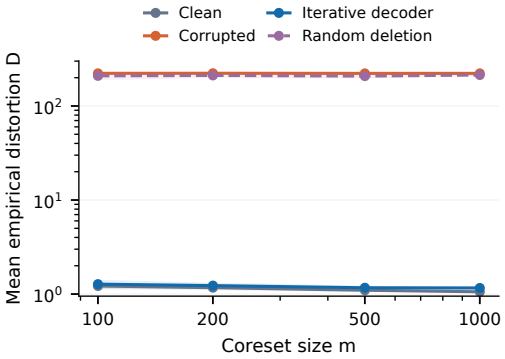


Figure 6: Distortion on CoverType under adversarial coordinate and weight corruptions. Same settings as Figure 4.

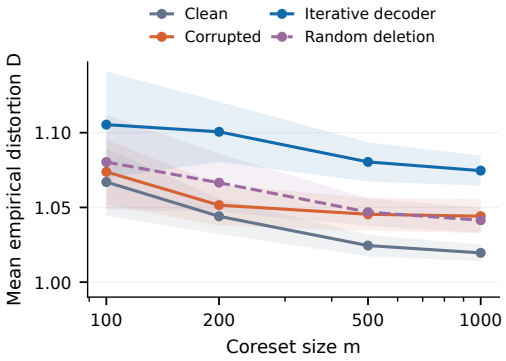


Figure 7: Distortion on MNIST under random Gaussian corruptions. Same settings as Figure 3.

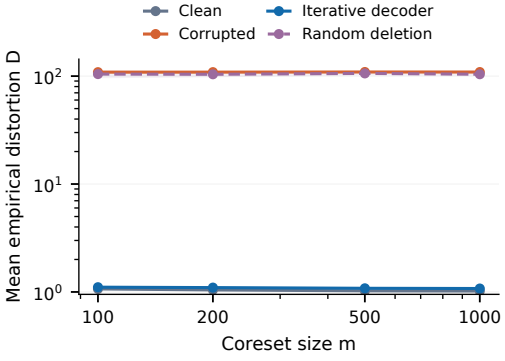


Figure 8: Distortion on MNIST under adversarial coordinate and weight corruptions. Same settings as Figure 4.

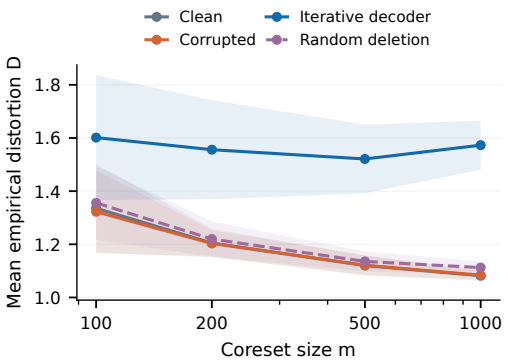


Figure 9: Distortion on Taxi under random Gaussian corruptions. Same settings as Figure 3.

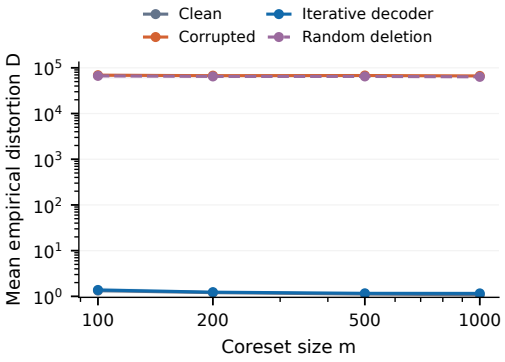


Figure 10: Distortion on Taxi under adversarial coordinate and weight corruptions. Same settings as Figure 4.

when $V_Q > 0$. We recompute these sensitivities after every deletion and evaluate distortion analytically over all Euclidean single centers. At $m = 1000$ and $f = 50$, mean decoded distortions range from 1.101 to 1.117. In a separate fixed-budget check with $f = 10$ and $m \in \{200, 500, 1000, 2000\}$, decoded distortion decreases as m grows; for Adult its mean decreases from 1.143 to 1.028.

For an exact two-center check, we use all 48,842 raw Adult ages and fix the public ambient domain

$$X = \{-200, -100\} \cup \{0, 1, \ldots, 100\} \cup \{200, 300\}.$$

Every pair of distinct points in X is a feasible query. The decoder receives this public domain, the corrupted summary, and f, but not clean ages or their frequencies. Sensitivities are recomputed exactly at every step, and evaluation covers all 5,460 center pairs. We use the same four sample sizes with $f = 0.05m$, plus a separate $m = 1000$, $f = 5$ check, with 20 trials per setting. Adversarial faults replace the largest-weight entries by random elements of $\{-200, -100, 200, 300\}$ and multiply their weights by 10. Random faults add Gaussian age noise of standard deviation 2, then round and clip to $[0, 100]$, keeping corruptions inside X.

At $m = 1000$ and $f = 5$, mean adversarial distortion falls from 46.875 to 1.072, compared with clean distortion 1.067 and random-deletion distortion 46.138. All five faults are removed in every trial. At $f = 50$, the corresponding corrupted and decoded distortions are 445.311 and 1.216. Under random age noise, the noisy and decoded distortions are 1.067 and 1.079 at $f = 5$, and 1.066 and 1.254 at $f = 50$. The exact sensitivity total is 3.499, giving the bound $fT/m = 0.0175$ at $f = 5$.

Finally, we test the need to recompute sensitivities on a nondegenerate masking example: 5,000 one-dimensional $\mathcal{N}(0, 0.01^2)$ points, $m = 1000$, and two entries corrupted to locations 1 and $M \in \{100, 10^4, 10^6\}$. Over 100 trials, at $M = 10^4$ one-shot removal gives mean exact distortion 11.112 and removes only one fault, whereas iterative removal gives 1.034 and removes both. Figure 12 illustrates the difference.

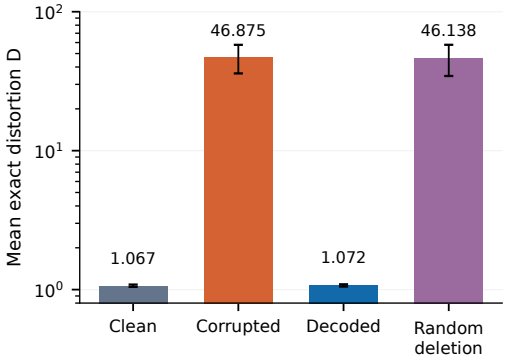


Figure 11: Exact Adult-age recovery, $k = 2$, $m = 1000$, $f = 5$, over all 5,460 public-domain queries. Means and standard deviations over 20 trials.

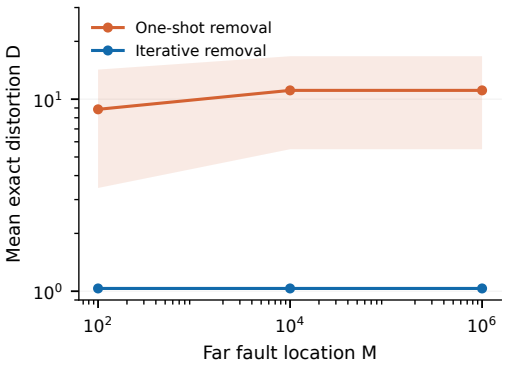


Figure 12: Exact one-center distortion under two masking faults. Means and standard deviations over 100 trials. Both axes are logarithmic.

C.2 Worst-Case Effect of Adversarial Corruptions on BICO and StreamKM++

BICO and StreamKM++ both are *movement-based* coresets. Such coresets use the property that a mapping π of point set P to another point set Q results in an $O(\varepsilon)$-coreset if $\sum_{p \in P} d^2(p, \pi(p)) \leq \varepsilon^2 \cdot OPT$, where OPT is the cost of an optimal k-means clustering. This mapping is typically used to summarize a large number of very close points by choosing the image to be the same point q and weighting q by the number of points in the pre-image of q. This property alone does not imply fault tolerance. For $k = 1$, consider a point set P consisting of n colocated points and a coreset Ω consisting of one of these points weighted by n, which would also be the summary generated by BICO and StreamKM++ on this instance. Moving the sole point in Ω to a different location corrupts the coreset with no possibility of recovery by filtering with inherited weights.

The goal of the experiment is to test whether adversarial corruption can cause cost underestimation that a filtering decoder cannot repair. Given a candidate clustering solution S, the adversary sets to zero the weights of the f coreset entries with largest weighted cost to S. This changes at most f entries and is admissible in our model, which permits arbitrary nonnegative replacement weights. Coordinates and all other weights remain unchanged, and the original dataset P is not modified.

Specifically, let $I_f(S)$ index the f largest values $w_i d(x_i, S)^2$ and let $\widetilde{\Omega}_{f,S}$ denote the resulting corrupted coreset. Its cost is

$$C_f(S) = \mathrm{cost}(\Omega, S) - \sum_{i \in I_f(S)} w_i d(x_i, S)^2.$$

For a common query family $\mathcal{Q}$, the quantity measured is the maximum witnessed distortion

$$D_f^{\mathrm{wit}} = \max_{S \in \mathcal{Q}} \max \left\{ \frac{C_f(S)}{\mathrm{cost}(P, S)}, \frac{\mathrm{cost}(P, S)}{C_f(S)} \right\}.$$

Each query can induce a different corruption; this is a maximum over query–corruption pairs, rather than evaluation of one common corrupted coreset on every query. Whenever $C_f(S) < \mathrm{cost}(P, S)$, every weighted submultiset $Q \subseteq \widetilde{\Omega}_{f,S}$ with inherited weights satisfies

$$\mathrm{cost}(Q, S) \leq C_f(S).$$

Consequently, the reciprocal ratio for that witness is a lower bound on the distortion of every permitted filtering decoder. We do not apply a solution-aware trimming decoder in this comparison. Sensitivities themselves are well-defined for movement-based summaries; the sampling-based recovery theorem does not automatically apply to their construction.

Experiment Details. We ran this test on the Taxi and MNIST datasets. For Taxi, we used a random subset of 300,000 endpoints, with dimension $d = 2$, and repeated the analysis on 300,000 geographically bounded endpoints. For MNIST, we used 10,000 images represented as vectors in $\mathbb{R}^{784}$. We

tested $k \in \{50, 100\}$ and corruption budgets

$$f \in \{0, 1, 2, 5, 10, 20, 40, 80\}.$$

Each case uses 20 construction trials on a fixed input subset. The primary BICO implementation follows the published insertion and rebuilding procedures with threshold doubling Fichtenberger et al. [2013]. StreamKM++ uses weighted TreeCoreset splitting and binary merge-and-reduce Ackermann et al. [2012]. These are fresh implementations from the published descriptions; optional nearest-neighbor accelerators are omitted. Sensitivity sampling uses the standard proxy below.

For each dataset and value of k, we first computed a high-quality reference clustering S_{ref} on the original dataset. The common query bank consists of k queries: each removes one reference center and inserts a remote unused center, retaining exactly k centers. It is fixed independently of the coreset draws and used for all methods. The adversary then chooses the largest weighted contributors separately for each query and method.

We use a 2,000-entry cap on Taxi and a 1,000-entry cap on MNIST. BICO's actual entry count is matched exactly by StreamKM++ and sensitivity sampling in each trial. Actual counts range from 1,754 to 1,850 on raw Taxi, 1,231 to 1,999 on geographically bounded Taxi, and 408 to 987 on MNIST. The $f = 0$ results are the clean controls. We report each corruption budget separately rather than averaging over different values of f.

Empirical Observations and Trends. Taxi provides a clear fault-specific separation. With $k = 50$ and $f = 5$, mean witnessed distortions are 1.689 for BICO, 1.686 for StreamKM++, and 1.124 for sensitivity sampling, compared with clean values 1.018, 1.004, and 1.091. With $k = 100$, the corresponding attacked distortions are 1.817, 1.770, and 1.299. The separation survives geographic restriction: for $k = 50$ and $f = 5$, mean distortions are 1.683, 1.708, and 1.132. This is consistent with concentrated movement summaries losing important represented mass when a few entries are erased. The sampling advantage narrows substantially at larger budgets; for example, at $k = 50$ and $f = 80$ on raw Taxi, the means are 2.218, 2.216, and 2.125.

MNIST is more sensitive to the clean approximation quality of the implementation. At $k = 50$, BICO's clean distortion is already 3.258, compared with 1.201 for StreamKM++ and 1.023 for sensitivity sampling; at $f = 20$, these become 4.249, 1.359, and 1.069. Thus BICO's large post-corruption MNIST error does not by itself isolate fault damage. A separate fixed-threshold BICO control, with entry counts again matched, gives the same qualitative Taxi separation; for $k = 50$ and $f = 5$, the means are 1.665, 1.669, and 1.116. Figure 13 shows the primary full-budget comparison.

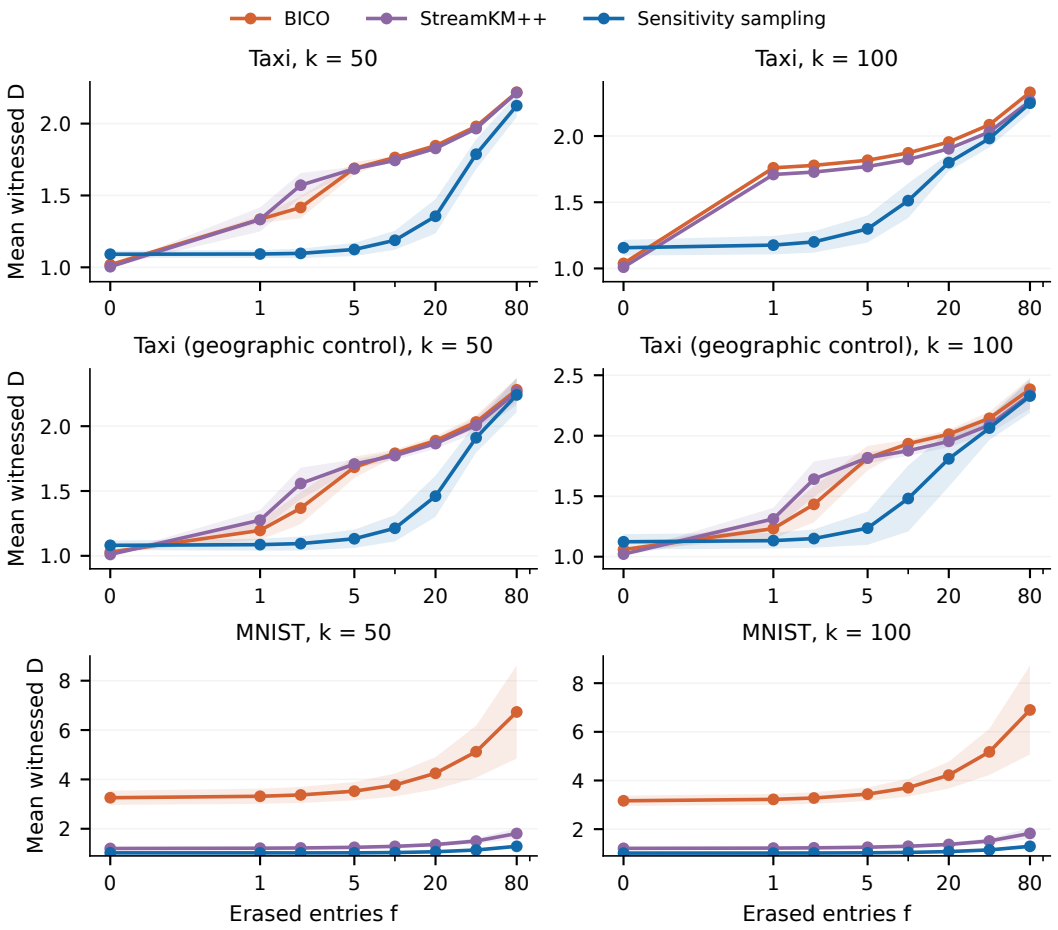


Figure 13: Comparison between BICO with rebuilding, StreamKM++ and sensitivity sampling under targeted weight erasures. Rows show raw Taxi, geographically bounded Taxi, and MNIST; columns show $k = 50$ and $k = 100$. Lines show means over 20 trials, shading indicates one standard deviation, and $f = 0$ is the clean control. Entry counts are matched within each trial.

C.3 Point-Wise Approximated Sensitivities vs Sensitivity Proxies

Third, we evaluate the practical implications of computing sensitivities accurately, which is also one of our contributions in Section 4. Many existing implementations rely on proxy upper bounds rather than computing sensitivities directly. We therefore compare coresets constructed using standard proxy sensitivities with coresets constructed using exact finite-domain sensitivities or practical sensitivity estimates. We perform this comparison on standard datasets as well as on a synthetic instance designed to highlight failure modes of the proxy sensitivity distribution.

As mentioned in Section A.1, the commonly understood algorithm referred to as sensitivity sampling uses slightly different probabilities than point-wise approximate sensitivities. The most powerful of these proxies also have optimal worst-case guarantees Bansal et al. [2024], Draganov et al. [2024] and are empirically competitive among previous coreset implementations Schwiegelshohn and Sheikh-Omar [2022]. Here, we investigate whether more accurate sensitivities offer an empirical improvement over a standard proxy, complementing the analysis of Schwiegelshohn and Sheikh-Omar [2022]. Our comparison concerns the proxy specified below, rather than every optimal or fast proxy variant.

This experiment compares four sampling distributions for k-means coresets on Adult, MNIST, CoverType, and Taxi: uniform sampling, lightweight coreset sampling Bachem et al. [2018], the

standard proxy sensitivity, and exact or estimated sensitivities. The proxy is proportional to

$$\tau_i = \frac{1}{|C(i)|} + \frac{\|x_i - c(i)\|^2}{\sum_j \|x_j - c(j)\|^2},$$

where $c(i)$ and $C(i)$ are the assigned reference center and cluster. Lightweight sampling mixes uniform probabilities with squared distance to the global mean in equal proportions. All scores are normalized to sampling probabilities. Each coreset is evaluated over the same fixed candidate center family for that dataset and setting. The goal is to see whether the sensitivity scores give lower distortion, lower variance, or smaller required coreset size than the usual proxy while keeping evaluation identical across methods.

Experimental Setup. For the practical ten-center experiment, the dataset is restricted to $n_{\exp} = 1000$ points, with $k = 10$, and coreset sizes

$$m \in \{100, 200, 500, 1000\}, \qquad m/k \in \{10, 20, 50, 100\}.$$

We use three independently sampled subsets per dataset and 200 draws per subset and size. The candidate center domain is 64 input points selected by D^2 sampling. For each input point, five radii restrict the admissible centers while leaving at least ten candidates; greedy center addition approximately solves each restricted problem. The resulting scores are feasible-solution lower estimates, not certified constant-factor approximations. Evaluation uses 128 independently generated ten-center subsets of the same domain. We plot mean distortion with standard-deviation bands and the 90th percentile of distortion against m on a logarithmic horizontal axis.

For exact scores, we use five independently sampled 1,000-point subsets per dataset and $k = 2$. The center domain is the set of distinct input locations, and we enumerate every pair of distinct candidate centers to compute sensitivities. Evaluation uses a separate family of 128 pairs, drawn from snapped local solutions, k-means++ seeds, random pairs, and distant pairs. Each setting uses 200 draws per subset and size. In an exhaustive audit, we use three fresh 200-point subsets per dataset, $m \in \{25, 50, 100\}$, and 250 draws per subset, evaluating both sensitivities and distortion over every one of the 19,900 center pairs. Exactness in these tests refers to the specified finite domain, not arbitrary Euclidean centers.

Empirical Observations and Trends. On real-world datasets, accurate scores improve over the chosen proxy in specific tested settings. On CoverType at $m = 100$, the practical $k = 10$ scores give mean empirical distortion 1.152, versus 1.274 for the proxy and 1.167 for lightweight sampling. The exact finite-domain $k = 2$ comparison gives 1.080 versus 1.115 for the proxy; lightweight sampling is essentially tied at 1.080. The proxy-minus-exact difference is positive on all five tested subsets, with mean 0.0355 and a conditional paired 95% bootstrap interval $[0.0325, 0.0385]$. This interval describes sampling variation within the tested subsets, not population-wide uncertainty.

The exhaustive 200-point audit retains favorable gaps: at $m = 50$, Adult gives mean distortion 1.159 for exact scores versus 1.208 for the proxy, and CoverType gives 1.160 versus 1.195. Lightweight sampling again performs comparably. Figures 14, 15, 16, and 17 show the practical ten-center results; Figure 18 gives the exact finite-domain comparisons. These experiments support specific improvements over the chosen proxy without establishing universal superiority over other samplers.

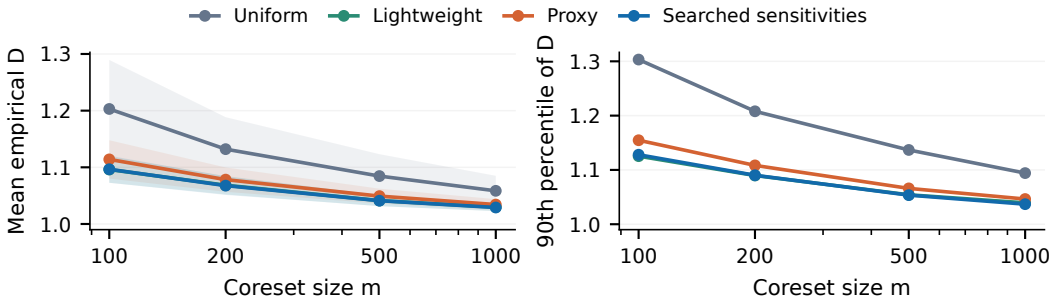


Figure 14: Comparison of sampling distributions on Adult, $k = 10$, with a 64-point candidate center domain. Left: mean empirical distortion with standard-deviation bands. Right: 90th percentile. Three subsets and 200 draws per subset; the searched sensitivities are heuristic estimates.

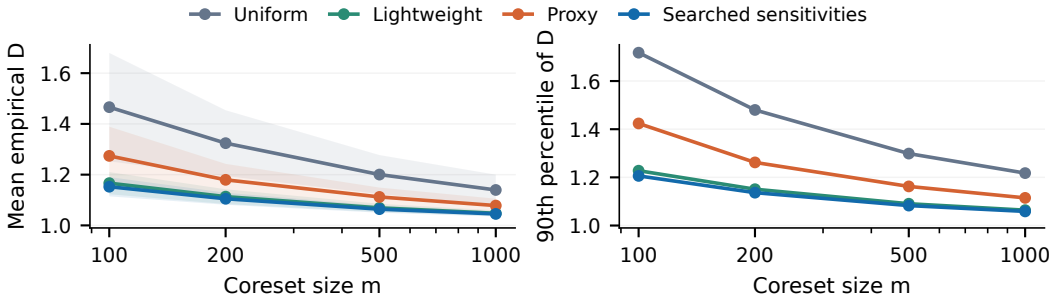


Figure 15: Comparison of sampling distributions on CoverType. Same settings and quantities as Figure 14.

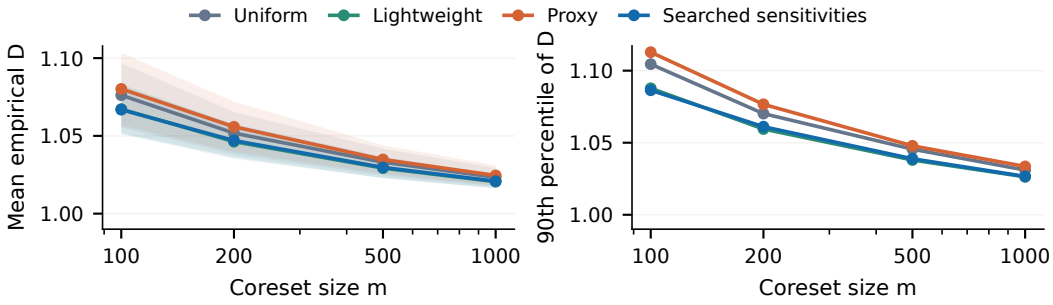


Figure 16: Comparison of sampling distributions on MNIST. Same settings and quantities as Figure 14.

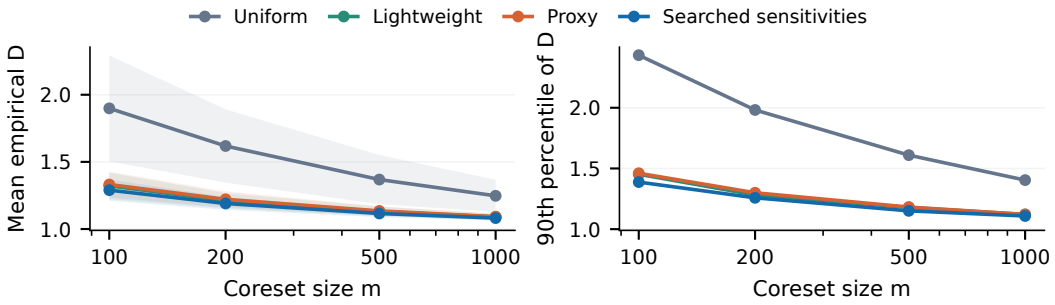


Figure 17: Comparison of sampling distributions on Taxi. Same settings and quantities as Figure 14.

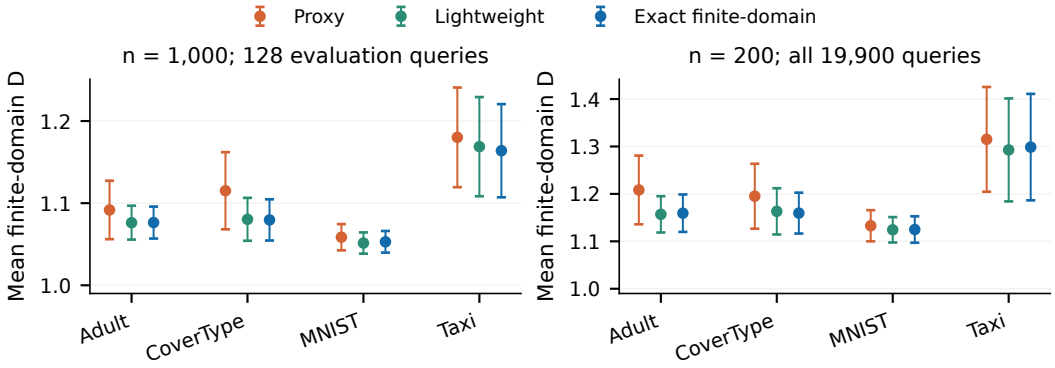


Figure 18: Exact finite-domain sensitivity comparisons. Left: $n = 1000$, $m = 100$, five subsets and 200 draws per subset, evaluated on 128 queries. Right: $n = 200$, $m = 50$, three subsets and 250 draws per subset, evaluated over all 19,900 queries. All queries have $k = 2$ distinct centers. Points show means; error bars show one standard deviation.

Description of the Synthetic Dataset. The simplex experiment is designed to expose the waste in the proxy sensitivity term $1/|C|$.

We take n vertices of a regular simplex, represented by the unit basis vectors $x_i = e_i$, so by symmetry the true sensitivity distribution is uniform. We fix a reference partition that makes the first $k-1$ vertices singleton clusters and puts all remaining vertices into one large cluster. This is an optimal k-means partition: a cluster of s vertices has centroid cost $s-1$, so every nonempty k-partition has total cost $n-k$. Thus the example does not require a bad bicriteria fit.

Writing $r = n-k+1$, the chosen proxy gives each singleton score 1 and each point in the large cluster score $2/r$. It therefore assigns total probability $2/(k+1)$ to the large cluster. To reveal the performance gap, we first choose k comparable to n, namely $n = 300$ and $k = 200$. The proxy spends most of its mass on the $k-1$ singleton vertices even though they are not more sensitive. Uniform and lightweight sampling coincide with normalized exact sensitivities on this symmetric instance.

The original witness is the center set

$$S = \{x_1, \ldots, x_{k-1}, 0\}.$$

The singleton vertices contribute zero cost to S, while the remaining r vertices determine its input cost. We include this witness and 63 fixed randomly selected basis-plus-origin queries, using

$$m \in \{100, 200, 500, 1000, 2000, 5000\}$$

and 10,000 draws per method and size. Because zero sampled cost gives infinite multiplicative distortion, Figures 19 and 20 plot the maximum absolute relative cost error over the 64 queries,

rather than labelling it as distortion. At $m = 2000$, mean maximum relative error is 0.0810 for exact/uniform sampling and 0.3185 for the proxy; 91.76% of exact/uniform draws put every query within 10% of its input cost, versus none for the proxy. At $m = 100$, 36.61% of proxy draws have an infinite-distortion witness. Sizes above $n = 300$ illustrate a sampling-distribution effect but are not compression evidence.

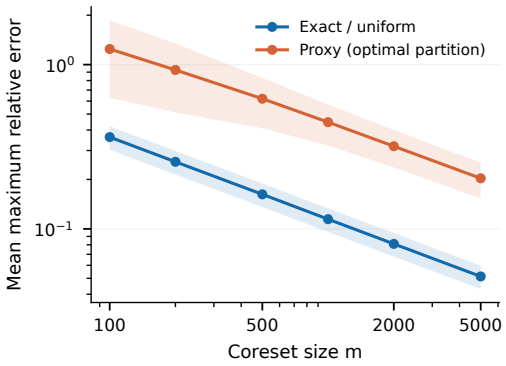


Figure 19: Mean maximum relative cost error on the small simplex, over 64 fixed queries and 10,000 draws per method and size. Bands show one standard deviation.

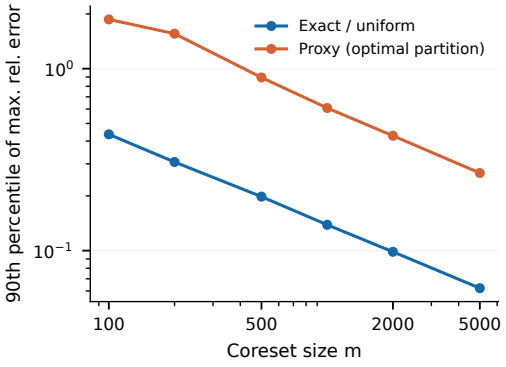


Figure 20: 90th percentile of maximum relative cost error on the small simplex. Same settings as Figure 19; these are relative-error statistics, not multiplicative distortion.

To test a compression regime, we also use $n = 30000$ and $k = 200$, with 2,000 draws per method and size and

$$m \in \{500, 1000, 2000, 5000, 10000\}.$$

We evaluate exactly over the entire fixed class

$$\mathcal{Q}_{\text{basis}} = \{\{0\} \cup \{e_i : i \in A\} : A \subseteq [n],\ |A| = k - 1\}.$$

Every query has input cost $r = n - k + 1$. If W_i is the total sampled weight at vertex i, including $W_i = 0$ for unsampled vertices, the minimum query cost is $\sum_i W_i$ minus the sum of the $k - 1$ largest W_i, and the maximum subtracts the $k - 1$ smallest instead. These order statistics give exact distortion over the whole class, without sampling center queries or storing a dense identity matrix. At $m = 5000$, median distortion is 1.084 for exact/uniform sampling and 804.033 for the proxy; all 2,000 exact/uniform draws satisfy $D \leq 1.1$, whereas none of the proxy draws do. At $m = 500$, 99.95% of proxy draws have infinite distortion. Figure 21 shows all tested sizes. This is an exact audit over the stated class, not a certificate over every Euclidean center set.

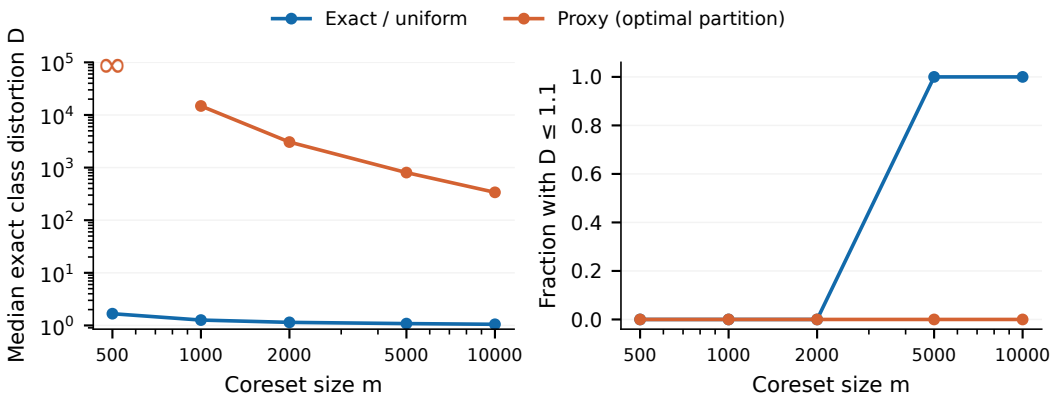


Figure 21: Compression-regime simplex, $n = 30000$, $k = 200$, and 2,000 draws per method and size. Left: median distortion, evaluated exactly over all basis-plus-origin queries; the infinity symbol records infinite median distortion. Right: fraction of draws with $D \leq 1.1$. All tested sizes are smaller than the input.

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: We provide both verbal and visual intuition as well as formal proofs of all claims stated in the paper. We also give an in-depth experimental evaluation of the proposed algorithms

   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 paragraph "Further comments and limitations" in Section 1.2 discusses settings outside the scope of the framework.

   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: The theoretical statements and proofs appear in Sections 3 and 4, with omitted proofs in Appendices A.1, A.2, and B.

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: We provide a full description of how to reproduce the experiments and include the code that is set up to automatically run all of them.

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: Section 5 links to the public code and experimental results.

   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 describes the evaluation methods; Appendix C gives dataset sizes, corruption parameters, coreset sizes, and experimental settings.

   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: We report means with one-standard-deviation bands or error bars for the main recovery, baseline, and real-data sampling-distribution comparisons. Trial and subset counts are specified in Appendix C. The exact finite-domain CoverType comparison also includes a paired 95% bootstrap confidence interval, conditional on the tested subsets. Percentile and success-frequency summaries are labelled separately.

   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:Appendix C reports the processor, RAM, software environment, thread limits, and approximate execution time for each experiment family. It also specifies the accounting of concurrent jobs and the additional research compute excluded from these estimates.

   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: We formulate and solve a theoretical problem which arises naturally from the literature.

   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: This work studies fault tolerance for coreset summaries and is not tied to a particular application or deployment. We do not identify a direct societal impact requiring separate discussion.

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: No such risk posed.

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: We use the Adult, Covertype, and MNIST datasets. Adult and Covertype are obtained from the UCI Machine Learning Repository and are licensed under CC BY 4.0. MNIST is credited to Yann LeCun and Corinna Cortes and is made available under CC BY-SA 3.0. We cite the original dataset sources and use the datasets only in ways consistent with their stated licenses and terms of use.

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: [Yes]

    Justification: Our code is well-documented and should reproduce experiments out of the box (once the datasets are downloaded).

    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: No crowdsourcing experiments were done.

    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: Not Applicable

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: Not applicable.

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.