# Sublinear Time Quantum Sensitivity Sampling

**Zhao Song**
Independent Researcher
magic.linuxkde@gmail.com

**David P. Woodruff**
Carnegie Mellon University and Google Research
dwoodruf@cs.cmu.edu

**Lichen Zhang**
Massachusetts Institute of Technology
lichenz@mit.edu

## Abstract

We present a unified framework for quantum sensitivity sampling, extending the advantages of quantum computing to a broad class of classical approximation problems. Our unified framework provides a streamlined approach for constructing coresets and offers significant runtime improvements in applications such as clustering, regression, and low-rank approximation. Throughout, we assume that $n$ and $d$ are polynomially larger than $k$ and $1/\epsilon$, and for regression that $n$ is polynomially larger than $d$. Running times report the leading term. Our contributions include:

- **$k$-median and $k$-means clustering:** For $n$ points in $d$-dimensional Euclidean space, we give an algorithm that constructs an $\epsilon$-coreset consisting only of points in the dataset, of size $\widetilde{O}(k^{4/3}\epsilon^{-2})$ for $k$-median and $\widetilde{O}(k^{3/2}\epsilon^{-2})$ for $k$-means, in time $\widetilde{O}(\epsilon^{-1}n^{0.5}dk^{1.67})$ for $k$-median and $\widetilde{O}(\epsilon^{-1}n^{0.5}dk^{1.75})$ for $k$-means. Our approach achieves a better dependence on $d$, with the running time depending only linearly on $1/\epsilon$, improving upon the main result of [Xue, Chen, Li and Jiang, ICML'23] in the high-dimensional regime $d \geq \Omega(k^{7/3})$ for $k$-median and $d \geq \Omega(k^{2.5})$ for $k$-means.
- **$\ell_p$ regression:** For $\min_{x\in\mathbb{R}^d} \|Ax - b\|_p$ where $A \in \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$, we construct an $\epsilon$-coreset of size $\widetilde{O}_p(d^{\max\{1,p/2\}}\epsilon^{-2})$ in time $\widetilde{O}_p(n^{0.5}d^{1.5}\epsilon^{-1})$ for $p \leq 2$ and $\widetilde{O}_p(n^{0.5}d^{p/4+1}(\epsilon^{-1} + d^{0.5}))$ for $p > 2$, matching [Liu and Ji, ICML'26] for $p \leq 2$ and improving upon the prior best quantum sampling approach of [Apers and Gribling, SICOMP'26] for every $p > 2$.
- **Low-rank approximation with Frobenius norm error:** We introduce the first quantum sublinear-time algorithm for low-rank approximation that approximates the best rank-$k$ solution to a matrix $A \in \mathbb{R}^{d\times n}$ and does not rely on data-dependent parameters, and, returning the factors in factored form, runs in $\widetilde{O}(n^{0.5}dk^{0.5}\epsilon^{-0.5})$ time. Materializing the factors explicitly costs an additional $\widetilde{O}(nk^2\epsilon^{-2})$ time. Additionally, we present quantum sublinear algorithms for kernel low-rank approximation and tensor low-rank approximation, broadening the range of achievable sublinear time algorithms in randomized numerical linear algebra.
- **Lower bounds:** We prove that any quantum algorithm for generic sensitivity sampling with total sensitivity $S$ that outputs an $\epsilon$-coreset of size $o(\sqrt{nS})$ must make $\Omega(\sqrt{nS})$ queries to the input, so the query complexity of our framework is optimal up to $\mathrm{poly}(\epsilon^{-1}, \log n)$ and VC-dimension factors.

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

# 1 Introduction

Given a set of points $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$, a universe $X$, and a cost function $\mathrm{cost} : \mathbb{R}^d \times X \to \mathbb{R}_{\geq 0}$, we study the problem of constructing a *coreset* of $A$: a weighted subset $B$ of points along with a nonnegative weight vector $w \in \mathbb{R}^{|B|}_{\geq 0}$ such that $\sum_{b \in B} w_b \cdot \mathrm{cost}(b, x) = (1 \pm \epsilon) \cdot \mathrm{cost}(A, x)$ for all $x \in X$, where $\mathrm{cost}(A, x) = \sum_{i=1}^{n} \mathrm{cost}(a_i, x)$. A coreset is particularly useful because it enables applying any existing approximation (or exact) algorithm on the smaller summary, yielding a good approximation to the original problem. Applications of coresets span clustering [Chen, 2009, Langberg and Schulman, 2010, Feldman and Langberg, 2011, Varadarajan and Xiao, 2012, Braverman et al., 2021a, Huang and Vishnoi, 2020, Braverman et al., 2021b, Cohen-Addad et al., 2021, 2022a,b, Huang et al., 2024], graph sparsification [Benczúr and Karger, 2015, Spielman and Teng, 2011, Spielman and Srivastava, 2011, Batson et al., 2012], hypergraph sparsification [Bansal et al., 2019, Kapralov et al., 2022, Jambulapati et al., 2023b, Lee, 2023], $\ell_p$ regression [Drineas et al., 2006, Clarkson, 2005, Dasgupta et al., 2009, Cohen and Peng, 2015, Woodruff and Yasuda, 2025a], submodular optimization [Rafiey and Yoshida, 2022, Jambulapati et al., 2023a], generalized linear models [Mai et al., 2021, Munteanu et al., 2022, Musco et al., 2022, Jambulapati et al., 2024], and subspace approximation [Cohen et al., 2015a, 2017, Woodruff and Yasuda, 2025b].

Coresets can be constructed via *sensitivity sampling*: the sensitivity of the $i$-th point is defined as $s_i = \max_{x \in X} \frac{\mathrm{cost}(a_i, x)}{\mathrm{cost}(A, x)}$. Sensitivity sampling draws point $i$ with probability proportional to $s_i$, reweighting each sampled point by the inverse of its sampling probability to ensure the estimator is unbiased. The seminal work of Langberg and Schulman [2010] shows that this yields a coreset with a simple and elegant proof: for any *fixed* $x \in X$, sampling proportional to sensitivity ensures low variance, and Bernstein's inequality implies that $O(\epsilon^{-2} S)$ samples suffice, where $S = \sum_{i=1}^{n} s_i$ is the *total sensitivity*. Recall that our goal is to approximate the cost for *all* $x \in X$. To do so, we union bound over a discretization of $X$ with size $\exp(\dim(X))$, where $\dim(X)$ is a notion akin to VC dimension [Feldman and Langberg, 2011]. Thus, a coreset can be constructed with $O(\epsilon^{-2} \cdot \dim(X) \cdot S \log(1/\delta))$ samples, where $\delta$ is the failure probability of the algorithm.

Algorithmically, a challenge arises: it is necessary to either compute or efficiently approximate $s_i$ or an upper bound on it. Most of the work in sensitivity sampling focuses on this task, and for many problems it can be achieved in nearly-linear time in $nd$ [Huang and Vishnoi, 2020, Spielman and Srivastava, 2011, Cohen and Peng, 2015, Cohen et al., 2017, Woodruff and Yasuda, 2025b]. Since scanning the entire dataset already takes $\Omega(nd)$ time, achieving nearly-linear time is close to optimal.

Typically, coresets are constructed for downstream *optimization problems*. For instance, coresets for $\ell_p$ regression help solve the original regression problem [Jambulapati et al., 2022, Adil et al., 2024], and coresets for subspace approximation yield column subset selection for low-rank approximation [Cohen et al., 2017]. In certain structured settings, some of these optimization problems admit *sublinear time* algorithms. For example, if the input matrix $A$ is positive semidefinite (PSD) [Musco and Woodruff, 2017, Bakshi et al., 2020] or Toeplitz [Musco and Sheth, 2024], one can obtain a rank-$k$ approximation in $n \cdot \mathrm{poly}(k/\epsilon)$ time, despite $A$ having size $n \times n$.

Can sensitivity sampling—and subsequently solving downstream optimization problems—be accomplished in sublinear time, even without structural assumptions? In this work, we explore this question through the lens of *quantum computing*, analyzing the time complexity of sensitivity sampling under quantum algorithms. Notably, tasks like linear regression, low-rank approximation, and clustering have quantum algorithms running in $o(nd)$ time [Kerenidis and Prakash, 2017, Kerenidis et al., 2019, Gilyén et al., 2022, Shah and Jaiswal, 2025], though these often rely on special input representations that support efficient weighted sampling. Moreover, their runtimes often depend on data-specific parameters such as $\|A\|_F$, condition number $\kappa(A) = \sigma_{\max}(A)/\sigma_{\min}(A)$, or dataset radius. In contrast, we seek quantum algorithms that (1) operate in sublinear time, (2) are *independent of input representation*, and (3) have *runtime independent of data-specific parameters*.

In this work, we provide a generic quantum algorithm applicable to *sensitivity sampling* in general. Throughout the paper, we use $s$ to denote the size of the coreset and let $\mathcal{T}_{\mathrm{sensitivity}}(s, X)$ represent the time to approximate one sensitivity over a set of $s$ points and universe $X$. Our algorithm runs in time $\widetilde{O}(\sqrt{ns}) \cdot \mathcal{T}_{\mathrm{sensitivity}}(s, X)$ and makes $\widetilde{O}(\sqrt{ns})$ queries to the points in $A$, which implies that as long as $s^{0.5} \cdot \mathcal{T}_{\mathrm{sensitivity}}(s, X) = o(n^{0.5} d)$, we achieve sublinear runtime. Moreover, our algorithm avoids dependence on data-specific parameters: the sample size and sensitivity approximation time depend

only on $n$, $d$, $1/\epsilon$, $1/\delta$, and other problem-related parameters (e.g., $k$ in clustering and low-rank approximation, or $p$ in $\ell_p$ regression). We summarize the main result in the following theorem.

**Theorem 1.1** (Informal version of Theorem C.3). *Let $A \in \mathbb{R}^{n\times d}$ and let $X$ be a universe. Then, there exists a quantum algorithm that constructs an $\epsilon$-coreset $C$ of expected size $s := \widetilde{O}(\epsilon^{-2} \cdot \dim(X) \cdot S\log(1/\delta))$ with probability at least $1-\delta$, where $\dim(X)$ is the dimension of the range space induced by $X$ (Definition C.5, a shatter-function exponent), $S$ is the total sensitivity and $\epsilon, \delta \in (0,1)$. Moreover, if there exists a classical oracle that can output a constant-factor two-sided approximation[1] to one sensitivity over a set of $s$ points and universe $X$ in $\mathcal{T}_{\mathrm{sensitivity}}(s, X)$ time, then the quantum algorithm can be implemented in time $\widetilde{O}(\sqrt{ns} \cdot \mathcal{T}_{\mathrm{sensitivity}}(s, X))$, up to the cost of preparing the oracle once per recursion level (Theorem C.3), and $\widetilde{O}(\sqrt{ns})$ queries to the points in $A$.*

We emphasize that the runtime statement is conditional on a fast sensitivity oracle: it is sublinear in the input size only when $s^{0.5} \cdot \mathcal{T}_{\mathrm{sensitivity}}(s, X) = o(n^{0.5}d)$. For the applications in this paper, the weights in question (leverage scores, ridge leverage scores, and Lewis weights) reduce to quadratic forms and admit $\widetilde{O}_p(d)$-time oracles via Johnson–Lindenstrauss sketching, so this condition holds, while $(k,p)$-clustering, whose sensitivities only admit one-sided approximations, is handled by a separate algorithm.

The query complexity of our framework is essentially optimal. In Section I, we prove that any quantum algorithm performing generic sensitivity sampling – outputting a coreset of size $o(\sqrt{nS})$ for a cost function with total sensitivity $S$ – must make $\Omega(\sqrt{nS})$ quantum queries (Theorem I.1), via a reduction from the strong direct product theorem for quantum search [Klauck et al., 2007], adapting Theorem 8.1 of Apers and Gribling [2026]. Since the coreset size of Theorem 1.1 satisfies $s = \widetilde{O}(\epsilon^{-2}\dim(X) \cdot S)$, the $\widetilde{O}(\sqrt{ns})$ query count of our framework matches this lower bound up to $\mathrm{poly}(\epsilon^{-1}, \log n)$ and VC-dimension factors.

Our approach is simple and general: it constructs the sample by uniformly subsampling half of the points, recursively computing approximate sensitivities on this subset, and then resampling based on these estimates. This scheme was first used for leverage score sampling [Cohen et al., 2015b] and in recent quantum linear programming algorithms [Apers and Gribling, 2026]. We extend this strategy to sensitivity sampling.

As a key application, we adapt our framework to solve the low-rank approximation problem. Given a matrix $A \in \mathbb{R}^{d\times n}$, the goal is to find matrices $U, V$ of rank $k$ such that $\|A - UV^\top\|_F^2 \le (1+\epsilon)\cdot\|A - A_k\|_F^2$, where $A_k$ is the best rank-$k$ approximation to $A$. We provide a quantum algorithm that constructs a column coreset of $A$, resulting in a low-rank approximation algorithm that runs in time $\widetilde{O}(n^{0.5}dk^{0.5}\epsilon^{-0.5})$ (Theorem D.8). We note a recent result by Chen et al. [2025], which provides a quantum algorithm for approximating the top-$k$ eigenvectors of a Hermitian matrix. Their method computes an orthonormal basis $W \in \mathbb{R}^{n\times k}$ such that $\|WW^\top - \sum_{i=1}^k v_i v_i^\top\| \le \epsilon$, where $v_i$ is the $i$-th eigenvector of the Hermitian input matrix, in time $\widetilde{O}(n^{1.5}k/(\epsilon\gamma))$, where $\gamma$ is the spectral gap between $\lambda_k$ and $\lambda_{k+1}$. While powerful, their method targets spectral norm error and depends on $\gamma^{-1}$. In contrast, our algorithm selects and reweights subsets of rows and columns and approximates the Frobenius-norm optimal rank-$k$ solution, with no dependence on $\gamma$. This makes our method suitable for downstream applications for small $\gamma$. Another related work [Chepurko et al., 2022] considers quantum-inspired classical algorithms for low-rank approximation, where they assume data structure access to $A$ that can sample its rows and entries proportional to their $\ell_2^2$ norms in $O(\log(nd))$ time. We note that their result also has data-dependent parameters such as condition number and the ratio $\frac{\|A\|_F^2}{\sigma_k(A)^2}$ in its running times, and our algorithm does not rely on such data structures at all. More generally, our recursion requires samples from sensitivity, Lewis-weight, or ridge-leverage distributions that are defined *adaptively* by the coreset constructed at the previous recursive level. Standard sample-and-query access provides samples from a fixed distribution, typically proportional to squared row or entry norms, and does not directly supply these adaptive target distributions. Constructing and maintaining them is precisely the difficult step. Two remarks make the comparison precise. First, sample-and-query access is built from the raw input in $O(\mathrm{nnz}(A))$ classical time [Chia et al., 2022], and the exact row norms it provides already require reading every nonzero entry. Hence,

[1] By a constant-factor two-sided approximation, we mean an estimate $\widetilde{s}_i$ satisfying $c\cdot s_i \le \widetilde{s}_i \le C\cdot s_i$ for absolute constants $c, C > 0$. The oracle receives a dataset of cardinality $s$ together with the universe $X$, and outputs such an estimate for a single sensitivity.

from raw input every sample-and-query algorithm costs $\Omega(\mathrm{nnz}(A)) = \Omega(nd)$ on dense $A$, whereas our low-rank algorithms (Corollary D.3, Theorem D.8) make $\widetilde{O}(n^{0.5} \cdot \mathrm{poly}(d, k, 1/\epsilon))$ column and row queries (Table 4). Our query lower bounds (Section I) are stated in the row query model, in which sample-and-query access is not provided, so they are unaffected. Second, given sample-and-query access, leverage score sampling (Theorem C.12) is reachable classically after oversampling by $\kappa^2(A)$ [Chepurko et al., 2022], and the column coreset of Corollary D.3 after oversampling by $\lambda/2$ with $\lambda = \|A\|_F^2/\|A - A_k\|_F^2$ [Chia et al., 2022]. In these cases the gain of our algorithms is the removal of these data-dependent factors and of the $\mathrm{nnz}(A)$ pass. We therefore do not expect a direct dequantization of our framework through this access model, although we do not claim a separation against all possible sample-and-query algorithms.

Beyond the recursive framework, we obtain new results for $(k, p)$-clustering, whose sensitivities admit only one-sided approximations and hence fall outside the framework. We instead give a quantum one-shot implementation of the coreset construction of Huang et al. [2024] seeded by a discrete constant-factor solution computed in $\widetilde{O}(n^{0.5}kd)$ time: the resulting algorithm outputs an $\epsilon$-coreset of input points only, of size $\widetilde{O}_p(k^{\frac{2p+2}{p+2}}\epsilon^{-2})$ as in Huang et al. [2024] up to polylogarithmic factors, in time $\widetilde{O}_p(\epsilon^{-1}n^{0.5}dk^{\frac{2p+3}{p+2}})$ (Theorem H.15). Compared with the recent quantum algorithm of Xue et al. [2023], whose main bound (their Theorem 3.1) is $\widetilde{O}_p(n^{0.5}d^{1.5}k^{0.5}\epsilon^{-(p/2\vee 1)})$ time with coresets of size linear in $d$ (their Johnson–Lindenstrauss variant, Remark 3.2 there, trades $d^{1.5}$ for $d$ at the price of $\max\{\epsilon^{-4}, \epsilon^{-p/2-3}\}$), our algorithm has a better dependence on $d$ than their main bound and no worse dependence on $1/\epsilon$, and it is faster than that bound, up to polylogarithmic factors, whenever $d \geq \Omega_p(k^{\frac{3p+4}{p+2}})$, i.e., $d \geq \Omega(k^{7/3})$ for $k$-median and $d \geq \Omega(k^{2.5})$ for $k$-means.

Our clustering algorithm has further implications for the *data selection pipeline* used in training foundation models. As discussed in Axiotis et al. [2024], if the loss function $\ell$ and all $k$-center solutions satisfy the $(p, \Lambda)$-well-behaved property, then a subset of $s = O(\epsilon^{-2})$ points suffices for training or fine-tuning: one first computes $k$ centers $x = (x_1, \ldots, x_k)$ using a clustering algorithm, then samples $s$ points according to the loss values $\ell(x_i)$. In our pipeline, the centers are computed from our coreset, and the second round of sampling requires at most $k$ queries to the loss function. Both steps run in $\widetilde{O}(n^{0.5}d \cdot \mathrm{poly}(k/\epsilon))$ time, whereas classical algorithms for this pipeline would require $\Omega(n)$ time. Our method is hence the first sublinear-time quantum algorithm for data selection pipelines.

We note that an independent and concurrent work of Liu and Ji [2026] studies quantum speedups for sparsifying generalized linear models (in turn they obtain improved quantum algorithms for various regression problems), via leverage score sampling over multiple scales [Jambulapati et al., 2024]. While one could interpret the resulting sparsified model obtained by them as a coreset for generalized linear models, their approach significantly differs from ours as we directly operate with approximate sensitivities or proxies to sensitivities, and our generic framework applies to generating any coreset. For $\ell_p$ regression with $p \in (0, 2]$, Liu and Ji [2026] obtain $\widetilde{O}(rn^{0.5}d^{0.5}\epsilon^{-1}) + \mathrm{poly}(d, 1/\epsilon)$ time for row sparsity $r$, and Theorems C.18 ($p < 2$) and C.12 ($p = 2$) match this on dense rows with $\widetilde{O}_p(n^{0.5}d^{1.5}\epsilon^{-1} + d^\omega)$ time. On $r$-sparse rows, our oracles cost only $\widetilde{O}_p(r)$ per query, so we also match their sparse bound (Remark C.19). We summarize our results in the following tables. Table 1 compares our coreset construction runtimes with prior work, and Table 2 compares runtimes for solving the corresponding optimization problems.

**Our contributions.** We summarize our main contributions below:

- We introduce a general quantum weighted sampling framework. Given weights $w \in \mathbb{R}^n_{\geq 0}$ satisfying mild conditions and access to a classical oracle that approximates the weight of a point over a small set, the framework constructs a coreset using $\widetilde{O}(\sqrt{ns})$ queries to the input, where $s$ is the size of the output coreset, and for an $\epsilon$-coreset, $s = \widetilde{O}(\epsilon^{-2} \cdot \mathrm{sum}(w))$ up to a dimension-dependent factor, where $\mathrm{sum}(w) \geq \sum_{i=1}^n w_i$ is an upper bound on the total weight. We show that sensitivity, leverage scores, and Lewis weights all meet these conditions, implying that coreset construction with these weights can be accelerated within our framework.
- We design the first sublinear time quantum algorithms for several fundamental low-rank approximation tasks: Frobenius-norm low-rank approximation, PSD and kernel low-rank approximation, and tensor low-rank approximation. Our algorithms are purely sampling-based, and avoid dependence on data-dependent parameters.

Table 1: Comparison of running times for quantum algorithms to construct an $\epsilon$-coreset for the respective problems. Here $n$ points lie in $\mathbb{R}^d$, $k$ is the number of centers, $p$ the norm exponent and $\epsilon$ the accuracy. We assume $n, d \gg \mathrm{poly}(k, 1/\epsilon)$ (and $n \gg \mathrm{poly}(d)$ for the regression rows) and ignore all dependencies on functions that only depend on $p$ for simplicity of presentation. The entry marked with † compares against the prior quantum Lewis-weight sampling approach. Our bound improves upon it for every $p > 2$. The entry marked with $\star$ compares against the time to compute the sparsifier of the concurrent work Liu and Ji [2026]. For simplicity both entries assume dense points, and for points with at most $r$ nonzero entries both become $n^{0.5}d^{0.5}r\epsilon^{-1}$ (Remark C.19). The clustering bounds are established in Theorem H.15 and the regression bounds in Theorems C.18 and C.12.

| | Reference | Previous | Ours |
|---|---|---|---|
| $k$-Median Clustering | Xue et al. [2023] | $n^{0.5}d^{1.5}k^{0.5}\epsilon^{-1}$ | $n^{0.5}dk^{1.67}\epsilon^{-1}$ |
| $k$-Means Clustering | Xue et al. [2023] | $n^{0.5}d^{1.5}k^{0.5}\epsilon^{-1}$ | $n^{0.5}dk^{1.75}\epsilon^{-1}$ |
| $(k,p)$-Clustering | Xue et al. [2023] | $n^{0.5}d^{1.5}k^{0.5}\epsilon^{-(p/2\vee 1)}$ | $n^{0.5}dk^{\frac{2p+3}{p+2}}\epsilon^{-1}$ |
| $\ell_{p>2}$ Regression † | Apers and Gribling [2026] | $n^{0.5}(d^7 + d^{p/4+3.5}\epsilon^{-1})$ | $n^{0.5}d^{p/4+1}(\epsilon^{-1} + d^{0.5})$ |
| $\ell_{p\leq 2}$ Regression $\star$ | Liu and Ji [2026] | $n^{0.5}d^{1.5}\epsilon^{-1}$ | $n^{0.5}d^{1.5}\epsilon^{-1}$ |

Table 2: Comparison of running times for low-rank and subspace approximation problems. We compare with the prior best classical algorithms, as quantum algorithms for these problems either have not been studied, assume a data-structure input, or have data-dependent parameters in the running time. We assume all matrices/tensors are dense and $n, d \gg \mathrm{poly}(k, 1/\epsilon)$ (and $n \gg \mathrm{poly}(d)$ for the Previous subspace entries), and ignore lower order terms. Low-rank factors are returned in factored form (Remark D.9). We use $\mathcal{T}_\mathsf{K}$ for the time to evaluate the kernel on two data points and $\omega$ for the matrix multiplication exponent. The tensor rows are $(2+\epsilon)$-approximations, versus $(1+\epsilon)$ in Song et al. [2019]. The row marked with ‡ omits, in both columns, a common factor $n^{O(\delta)}$ and, in ours, the hypotheses of Theorem G.7. The row marked with § assumes the flatness precondition of Theorem F.2.

| | Reference | Previous | Ours |
|---|---|---|---|
| Low-Rank | Cohen et al. [2017] | $nd$ | $n^{0.5}dk^{0.5}\epsilon^{-0.5}$ |
| PSD Low-Rank | Bakshi et al. [2020] | $nk^{\omega-1}\epsilon^{-(\omega-1)}$ | $n^{0.75}k^{2.25}\epsilon^{-2.25}$ |
| Kernel Low-Rank | Bakshi et al. [2020] | $nk\epsilon^{-1}(\mathcal{T}_\mathsf{K} + k^{\omega-2}\epsilon^{2-\omega})$ | $n^{0.75}k^{1.25}\epsilon^{-1.25}(\mathcal{T}_\mathsf{K} + k\epsilon^{-1})$ |
| Tensor, Rank-$k$ ‡ | Song et al. [2019] | $n^3 + 2^{O(k^2\epsilon^{-1})}$ | $n^2k^2\epsilon^{-2} + 2^{O(k^2\epsilon^{-1})}$ |
| Tensor, Bicriteria | Song et al. [2019] | $n^3$ | $n^2k^{0.5}\epsilon^{-1}$ |
| Subspace, $1 \leq p < 2$ § | Woodruff and Yasuda [2025b] | $nd$ | $n^{1-p/4}dk^{p/4}\epsilon^{-1}$ |
| Subspace, $p \geq 2$ | Woodruff and Yasuda [2025b] | $nd$ | $n^{1-1/p}dk^{0.5}\epsilon^{-2/p}$ |

- We develop improved quantum algorithms for $(k,p)$-clustering in the high-dimensional regime $d \geq \Omega_p(k^{\frac{3p+4}{p+2}})$: we construct an $\epsilon$-coreset of size $\widetilde{O}_p(k^{\frac{2p+2}{p+2}}\epsilon^{-2})$, consisting of input points only, in time $\widetilde{O}_p(\epsilon^{-1}n^{0.5}dk^{\frac{2p+3}{p+2}})$ (Theorem H.15), and further demonstrate how our framework can accelerate data selection pipelines for training foundation models.
- We complement our algorithms with an $\Omega(\sqrt{nS})$ query lower bound for generic sensitivity sampling with total sensitivity $S$ for coresets of size $o(\sqrt{nS})$ (Theorem I.1), showing that our framework is query-optimal up to $\mathrm{poly}(\epsilon^{-1}, \log n)$ and VC-dimension factors.

We also clarify the relationship to prior work: Grover-based quantum sampling and the recursive halving scheme for leverage scores originate in prior work (see Sections 2 and A.1). Our contribution lies in the sufficient conditions (Assumption C.1 and Theorem C.3) that make recursive quantum weighted sampling applicable in general, the proof that generic sensitivities satisfy these conditions, the stronger approximator formulations required for Lewis weights and ridge leverage scores, and the problem-specific classical reductions for the low-rank, kernel, and tensor variants.

**Roadmap.** Section 2 gives a technical overview of the framework, its instantiations, the low-rank and clustering algorithms, and the lower bounds, and Section 3 discusses open problems. The appendix contains the full proofs. Appendix A continues the overview, Appendix B collects preliminaries, and Appendix C proves the framework and its instantiations with leverage scores, Lewis weights, and ridge leverage scores. Appendices D, E, F, G, and H treat low-rank approximation, kernel low-rank

approximation, subspace approximation, tensor low-rank approximation, and clustering. Appendix I proves the lower bounds, Appendix J explains how the classical oracles are run in the quantum model, and Appendix K lists the query complexities.

**Quantum computation model.** We adopt the standard quantum computation model used in, e.g., Apers and de Wolf [2022], Apers and Gribling [2026]. This model supports quantum subroutines operating on $O(\log n)$ qubits, allows quantum queries to the input, and grants access to a quantum-read/classical-write RAM (QRAM) of size $\mathrm{poly}(n)$ bits. Each quantum read or classical write to QRAM incurs unit cost. We measure *time complexity* by the number of QRAM operations, and *query complexity* by the number of input queries made by the algorithm. QRAM is a common model studied in quantum algorithms. However, it is known that practically realizing the QRAM architecture is challenging. Throughout the paper, we often refer to randomized, classical routines as classical oracles. Converting these routines into quantum circuits would require one to analyze the complexity of cleaning up the garbage states. We circumvent the issue by adopting a derandomization approach due to Apers and de Wolf [2022] that "simulates" access to a random oracle via a $k$-wise independent hash function. Specifically, we use the following result of Apers and de Wolf [2022]:

**Claim 1.2** (Claim 1 in Apers and de Wolf [2022] and Lemma 3.12 of Apers and Gribling [2026])**.** *Consider any quantum algorithm with runtime $q$ that makes queries to a uniformly random string. We can simulate this algorithm with a quantum algorithm with runtime $\widetilde{O}(q)$ without random string, using an additional QRAM of $\widetilde{O}(q)$ bits.*

Given this claim, we can safely simulate any classical procedure without randomness. We refer to Section J for a more detailed discussion. We also note that the query complexities we obtained in this work are *independent* of QRAM, and they remain sublinear in $n$. We defer the details to Section K.

## 2 Technical Overview

Our entire framework can be stated in one sentence: *recursively halve the dataset, use a coreset of the half to estimate every point's sampling weight up to a constant factor, Grover-sample against these estimates with $\widetilde{O}(\sqrt{ns})$ queries, and recurse to depth* $\log n$. Concretely, the pipeline is: (i) uniformly subsample half of the points, $A' \subset_{1/2} A$, (ii) recursively construct a coreset (or approximator) $C'$ of $A'$, (iii) use $C'$ to implement a cheap classical oracle for approximate weights of the points of $A$, and (iv) Grover-sample against these estimates using $\widetilde{O}(\sqrt{ns})$ queries (Lemma B.15) and reweight, obtaining a coreset of $A$. The subsections below and Appendix A expand each ingredient and the applications. Table 3 summarizes the weight notions we instantiate the framework with, the approximation guarantee each requires, and the per-query oracle cost that makes step (iii) fast.

Table 3: The weight notions used in this paper, the approximation guarantee each requires for the recursion, and the per-query oracle cost. Clustering sensitivities admit only one-sided approximations, so they are handled by a separate, non-recursive algorithm (Section 2.5).

| Weight | Approximation notion | Oracle time | Result |
|---|---|---|---|
| generic sensitivity | $\epsilon$-coreset | $\mathcal{T}_{\mathrm{sensitivity}}(s, X)$ | Theorem C.3 |
| leverage score | $\epsilon$-coreset (spectral) | $\widetilde{O}(d)$ | recovers Apers and Gribling [2026] |
| $\ell_p$ Lewis weight | $\epsilon$-approximator (weights) | $\widetilde{O}_p(d)$ | Theorem C.18 |
| ridge leverage score | $\epsilon$-approximator (add.-mult.) | $\widetilde{O}(d)$ | Corollary D.3 |
| clustering sensitivity | one-sided only | $O(kd)$ | Theorem H.15 |

### 2.1 A Generic Recursive Sampling Framework

Our framework rests on a single quantum primitive, a sampling procedure based on Grover search [Grover, 1996]: associate each item with a value $q_i \in [0, 1]$, and the goal is to sample a subset where the $i$-th item is included with probability $q_i$. Given $q_i$'s, we can sample the subset in $\widetilde{O}(\sqrt{n \sum_{i=1}^{n} q_i})$ time, and if $\sum_{i=1}^{n} q_i = o(n)$, this is sublinear in $n$. The procedure requires an oracle that returns the value of $q_i$ upon query (see Lemma B.15 for a formal statement). Grover-based speedups of this kind have previously been obtained for edit distance, Laplacian solving, and linear

programming [Boroujeni et al., 2021, Gibney et al., 2024, Apers and de Wolf, 2022, Apers and Gribling, 2026].

The difficulty is that the weights we wish to sample from depend on the entire dataset, so the oracle itself is expensive to implement directly. Our framework resolves this with a recursive scheme governed by three conditions. Let $w(A, B) \in \mathbb{R}_{\geq 0}^{|A|}$ denote generalized weights of $A$ measured with respect to a reference set $B$, and say $B$ is an *$\epsilon$-approximator* of $A$ if for any $C$ and any point $c_i$ of $C$, we have $w_i(C, B) = (1 \pm \epsilon) w_i(C, A)$ (for sensitivities, an $\epsilon$-coreset plays this role). Three conditions suffice:

- **(C1) Consistent total weights:** For any subset $S \subseteq [n]$, $\sum_{i \in S} w_i(A_S, A_S) \leq \mathrm{sum}(w)$, where $\mathrm{sum}(w)$ is a finite upper bound on the total weight of any subset measured with respect to itself (for sensitivities, the total sensitivity). *Role:* (C1) bounds the total weight of every recursive subproblem, in particular of the uniform half and of the sets the oracle is built from, and hence the expected number of samples at every level. A level that draws $s$ samples in expectation costs $\widetilde{O}(\sqrt{ns})$ Grover queries.
- **(C2) Uniform sampling bound:** Let $A' \subset_{1/2} A$ be the subset that contains each point of $A$ independently with probability $1/2$. For a weighted set $B$, define the augmented weights $w'_i(A, A'; B)$ as $w_i(A, B)$ if $a_i \in A'$ and $w_i(A, B \cup \{a_i\})$ otherwise, and write $w'_i(A, A') = w'_i(A, A'; A')$. Then $w'_i(A, A') \geq w_i(A, A)$ for all $i \in [n]$. *Role:* (C2) guarantees that weights computed from the halved dataset never underestimate. The oracle of the recursion evaluates $w'_i(A, A'; C')$ for an approximator $C'$ of $A'$, which is within $1 \pm \epsilon$ of $w'_i(A, A')$ (for $a_i \in A'$ because $C'$ approximates $A'$, and for $a_i \notin A'$ because $C' \cup \{a_i\}$ approximates $A' \cup \{a_i\}$), so the recursion remains sound.
- **(C3) Importance sampling bound:** Let $u_i \geq w_i(A, A)$ be overestimates of the weights. Suppose we sample according to $q_i = \min\{1, \alpha u_i \log(1/\delta)\}$ for a large enough $\alpha$, and reweight the sample by $1/q_i$, then with probability at least $1 - \delta$, the weighted sample is an $\epsilon$-approximator of $A$ of size $O(\alpha \cdot \|u\|_1 \log(1/\delta))$, which is $O(\alpha \cdot \mathrm{sum}(w) \log(1/\delta))$ in expectation for the overestimates the recursion produces, by (C1). *Role:* (C3) shows that resampling against overestimates yields a valid approximator, closing the recursion.

Let $s = O(\alpha \cdot \mathrm{sum}(w) \log(1/\delta))$. Together, (C1)–(C3) drive Algorithm 1: line 5 halves the dataset, line 7 recurses, line 8 implements the weight oracle $w'_i(A, A'; C')$ from the child coreset $C'$, whose estimates are valid overestimates (up to the factor $1 - \epsilon$, absorbed by the constant 1000) by (C2), and line 9 Grover-samples with $\widetilde{O}(\sqrt{ns})$ expected queries by (C1), since the expected number of samples is $O(\alpha \cdot \mathrm{sum}(w) \log(1/\delta)) = O(s)$. Correctness of the resample follows from (C3). We obtain an algorithm that computes an $\epsilon$-coreset in the desired $\widetilde{O}(\sqrt{ns}) \cdot \mathcal{T}_{\mathrm{sensitivity}}(s, X)$ time (Theorem 1.1, formally Theorem C.3). We summarize an informal version of the algorithm below, where we use QSAMPLE to denote the Grover search procedure for sampling from a vector $q$ of probabilities (Lemma B.15).

**Algorithm 1** Quantum recursive sampling for coreset.

1: **procedure** QRECURSESAMPLE($A \in \mathbb{R}^{n \times d}, \alpha, \delta$)
2: **if** $n \leq \alpha \cdot \mathrm{sum}(w) \log(1/\delta)$ **then**
3: **return** $(A, I_n)$
4: **end if**
5: $A' \subset_{1/2} A$ ▷ Sample uniformly half of the rows from $A$
6: $s \leftarrow \alpha \cdot \mathrm{sum}(w) \log(1/\delta)$
7: $(C', D') \leftarrow$ QRECURSESAMPLE($A', \alpha, \delta$)
8: Implement a classical oracle for $w'_i(A, A'; C')$
9: $D \leftarrow$ QSAMPLE($q$) ▷ $q_i = \min\{1, 1000\alpha\, w'_i(A, A'; C') \log(1/\delta)\}$, failure parameter as in Algorithm 2
10: $C \leftarrow D^\top A$
11: **return** $(C, D)$
12: **end procedure**

### 2.2 Instantiations: Sensitivity and Lewis Weight Sampling

**Generic sensitivities.** We first instantiate the framework with generic sensitivities, taking $w_i(A, B) = s_i(A, B)$. The interesting condition to verify is the uniform sampling bound (C2). Consider the following simplified algorithm: form $A'$ by sampling each point of $A$ with probability $1/2$ uniformly, and define the generalized sensitivity as $s_i(A, A') = \max_{x \in X,\, \mathrm{cost}(A',x) \neq 0} \frac{\mathrm{cost}(a_i,x)}{\mathrm{cost}(A',x)}$, i.e., we change the denominator to $\mathrm{cost}(A', x)$. Note that this is *not* necessarily an overestimate of $s_i$: if the maximizer $x^*$ for $s_i$ has $\mathrm{cost}(A', x^*) = 0$, then $s_i(A, A')$ may fall below $s_i$. Including the point itself repairs this: $s_i(A, A' \cup \{a_i\}) \geq s_i$ always holds, since the denominator either stays positive and only shrinks relative to $\mathrm{cost}(A, x^*)$, or degenerates to $\mathrm{cost}(a_i, x^*)$, in which case the ratio equals 1, an upper bound on any sensitivity (a short case analysis in the proof of Theorem C.7 verifies this). We note that $s_i(A, A' \cup \{a_i\})$ is in fact the overestimate used by Cohen et al. [2015b] to obtain their initial uniform sampling bound.

The recursive framework follows directly: uniformly sample half of the points $A'$, then compute a coreset of $A'$, called $C$. To compute the overestimates of $s_i$, we use $s_i(A, C)$ for $a_i \in A'$ and $s_i(A, C \cup \{a_i\})$ for $a_i \notin A'$ (the augmented weights $w'_i(A, A'; C)$ of Section C with $w = s$), which is efficient since $C$ is a small-size coreset. Note that these are valid approximations of $s_i(A, A')$ and of $s_i(A, A' \cup \{a_i\})$, respectively – as $C$ is a coreset of $A'$, it approximates the cost of $A'$ with respect to all $x \in X$, and they have the same kernel (given a set of points $A$ and a cost function, the kernel of $A$ is $\ker(A) = \{x \in X : \mathrm{cost}(A, x) = 0\}$). Moreover, for $a_i \notin A'$, it is not hard to see that $C \cup \{a_i\}$ is also a coreset of $A' \cup \{a_i\}$ (Lemma C.2), and thus the sensitivity is preserved in both cases. To summarize, in each round of recursion, we are given a size-$s$ coreset, and assuming we can approximate each $s_i(A, C)$ or $s_i(A, C \cup \{a_i\})$ in $\mathcal{T}_{\mathrm{sensitivity}}(s, X)$ time, then the overall runtime is $\widetilde{O}(\sqrt{ns}) \cdot \mathcal{T}_{\mathrm{sensitivity}}(s, X)$, with recursion depth at most $\log n$ as we halve the points at each step, giving the desired runtime for sensitivity sampling.

**Lewis weights.** While the framework above handles *all* sensitivity sampling, in many applications, the exact sensitivities can be difficult to compute, and thus proxies are often sought as efficient alternatives. Take the $\ell_p$ regression problem as an example, where the sensitivity is $s_i = \max_{x \in \mathbb{R}^d, Ax \neq 0} \frac{|a_i^\top x|^p}{\|Ax\|_p^p}$. For $p = 2$, this corresponds to the leverage score, which can be quickly approximated. However, for general $p$, this is more complex, and algorithms for $\ell_p$ sensitivities tend to be less efficient than those for leverage scores [Padmanabhan et al., 2023]. Instead, constructing a coreset for $\ell_p$ regression is typically done *not* via sensitivity sampling, but through *Lewis weights sampling* [Bourgain et al., 1989, Ledoux and Talagrand, 1991, Talagrand, 1995, Schechtman and Zvavitch, 2001, Cohen and Peng, 2015, Woodruff and Yasuda, 2025a]. These weights are defined as the fixed-point solution for the following equation: $w_i^{2/p} = a_i^\top (A^\top W^{1-2/p} A)^{-1} a_i$, where $w_i$ represents the $i$-th leverage score of the matrix $W^{1/2-1/p}A$. Lewis weights have several desirable properties, such as $\sum_{i=1}^n w_i = d$, and they upper bound the $\ell_p$ sensitivities, exactly for $p \leq 2$ and up to a factor $d^{p/2-1}$ for $p > 2$. Moreover, Lewis weights can be approximated in nearly-linear time [Cohen and Peng, 2015, Lee, 2016, Jambulapati et al., 2022, Fazel et al., 2022, Apers et al., 2024, Gribling et al., 2026].

We defer the instantiation with ridge leverage scores, and the adaptation of our framework to Lewis weights, to Section A.1.

### 2.3 Optimality of the Framework: Query Lower Bounds

The $\widetilde{O}(\sqrt{ns})$ query count of the framework cannot be improved in its dependence on $n$ and $S$, and the obstruction is generic sensitivity sampling itself rather than any one application. Consider $n$ points in $\mathbb{R}^2$ split into $S$ blocks, where each point carries a hidden bit that one query reveals, and a point is marked if its bit is 1. The cost of a point under the $l$-th element of the universe is 1 if the point is a marked point of block $l$ and 0 otherwise. A marked point of a block has sensitivity equal to the inverse of the number of marked points in its block, and an unmarked point has sensitivity 0, so the total sensitivity is at most $S$ and the dimension $\dim(X)$ of the range space (Definition C.5) is $O(1)$. An $\epsilon$-coreset $B$ with $\epsilon < 1/2$ must contain a marked point of every block that contains a marked point, so reading the bits of its points, with at most $|B|$ further queries, decides for every block whether it contains a marked point. Computing these $S$ disjunctions at once costs $\Omega(S\sqrt{n/S}) = \Omega(\sqrt{nS})$

bounded-error quantum queries by the strong direct product theorem for OR [Klauck et al., 2007]. Hence, any quantum algorithm that outputs an $\epsilon$-coreset of size $o(\sqrt{nS})$ with probability at least $2/3$ makes $\Omega(\sqrt{nS})$ queries to the points (Theorem I.1). On this family the sensitivity oracle is trivially implementable, and Theorem C.3 produces a coreset of size $\widetilde{O}(\epsilon^{-2}S)$, which is $o(\sqrt{nS})$ whenever $S = \widetilde{o}(\epsilon^4 n)$, with $\widetilde{O}(\epsilon^{-1}\sqrt{nS})$ queries. Thus, the framework is query-optimal in $n$ and $S$ up to $\mathrm{poly}(\epsilon^{-1}, \log n)$ factors.

The dependence on $\epsilon$ is settled on the same family, under the hypotheses of Theorems I.3 and I.4. For $S = 1$ the weights of the coreset, together with the bits of its points, determine the number of marked points within a factor $1 \pm \epsilon$, which is approximate counting. Hence, the counting lower bound of Nayak and Wu [1999] shows that any quantum algorithm that outputs an $\epsilon$-coreset of size $o(\sqrt{n/\epsilon})$ with its weights makes $\Omega(\sqrt{n/\epsilon})$ queries. The counting algorithms and Grover search of Brassard et al. [2002] output an $\epsilon$-coreset of size 1 with $O(\sqrt{n/\epsilon})$ queries, so the bound is tight on this family (Theorem I.3). For general $S$, each block carries the partial function of Nayak and Wu [1999] that distinguishes two consecutive counts of marked points, whose query complexity is $\Omega(\sqrt{n/(S\epsilon)})$. The direct product theorem of Lee and Roland [2013] for partial functions then gives $\Omega(\sqrt{nS/\epsilon})$ queries (cf. Xue et al. [2023]) for any algorithm that outputs an $\epsilon$-coreset of size $o(\sqrt{nS/\epsilon})$ with its weights. Running the $S = 1$ algorithm on each block gives $O(\sqrt{nS/\epsilon}\log S)$ queries (Theorem I.4). Thus, the answer on the block family is $\Theta(\sqrt{nS/\epsilon})$ up to a $\log S$ factor, against the $\widetilde{O}(\epsilon^{-1}\sqrt{nS})$ of the framework, a gap of $\epsilon^{-1/2}$ up to logarithmic factors. For leverage scores ($S = d$), Lemma 8.2 of Apers and Gribling [2026] gives $\widetilde{\Omega}(\epsilon^{-1}\sqrt{nd})$ row queries, attained by Theorem C.12, when $n = O(d^2)$ and $\epsilon = \Omega(\sqrt{d/n})$, which force $S = \Omega(\epsilon^{-2})$. Whether the framework's $\epsilon^{-1}$ factor is necessary when $S = o(\epsilon^{-2})$ remains open. The bounds count queries to the points only.

### 2.4 Pure-Sampling Framework For Low-Rank Approximation

Given $A \in \mathbb{R}^{d\times n}$, the rank-$k$ low-rank approximation problem seeks to find a pair of matrices $U \in \mathbb{R}^{d\times k}$, $V \in \mathbb{R}^{n\times k}$ such that $\|A - UV^\top\|_F^2 \le (1+\epsilon)\|A - A_k\|_F^2$, where $A_k$ is the best rank-$k$ approximation of $A$. Low-rank approximation is closely related to the $(k, 2)$-subspace approximation coreset: let $\mathcal{F}_k$ be the set of all $k$-dimensional subspaces in $\mathbb{R}^d$, and define $\mathrm{cost}(a_i, x) = \|(I - P_x)a_i\|_2^2$ where $P_x$ is the orthogonal projection onto $x \in \mathcal{F}_k$. If we obtain a coreset $C$ for $A$, then we have for any $k$-dimensional orthogonal projection $P_x$, $\|(I - P_x)C\|_F^2 = (1 \pm \epsilon)\|(I - P_x)A\|_F^2$, which is sufficient to show that choosing $P_x$ as the projection onto $C_k$ will give the desired low-rank approximation [Cohen et al., 2017]. Moreover, instead of $(k, 2)$-subspace sensitivities, one could sample according to the *ridge leverage scores*, which can be computed quickly. To adapt our weighted sampling framework, we need to identify the right notion of an $\epsilon$-approximator for ridge leverage scores: a coreset $C$ of $A$ that additionally satisfies $(1-\epsilon)CC^\top - \epsilon\lambda_{A_k} I \preceq AA^\top \preceq (1+\epsilon)CC^\top + \epsilon\lambda_{A_k} I$, where $\lambda_{A_k} = \|A - A_k\|_F^2/k$. Thus, our framework gives a column coreset for the $k$-subspace cost in $\widetilde{O}(n^{0.5}dk^{0.5}\epsilon^{-1})$ time (Corollary D.3). For the low-rank approximation itself, the $\epsilon^{-1}$-oversampled column subset of Lemma D.5 brings the time to $\widetilde{O}(n^{0.5}dk^{0.5}\epsilon^{-0.5})$ (Theorem D.8, where the first term leads under the standing assumption $d \gg \mathrm{poly}(k, 1/\epsilon)$).

We defer the recovery of the solution through the subsampled columns to Section A.2, and the extension to tensor low-rank approximation via approximate regression by sampling responses to Section A.3.

### 2.5 Improved Coreset for Clustering with Applications

We also design an improved quantum algorithm for constructing an $\epsilon$-coreset of $(k, p)$-clustering. In contrast to the recursive sampling framework we developed in the preceding discussions, our algorithm is a quantum one-shot implementation of Huang et al. [2024], where the idea is to first compute a set of approximate $k$-centers, then perform one round of importance sampling on top of it. Why could our recursive sampling framework not be applied here? This is because the sensitivities of $(k, p)$-clustering can only be *overestimated*, and these overestimates in general do not satisfy the uniform sampling bound. In fact, a closer examination of our analysis shows that during the intermediate stages in the recursive sampling, we would need the sensitivities to be approximated in

a *two-sided* fashion, i.e., let $s_i$ be the exact sensitivities. We require the approximate sensitivities $\widetilde{s}_i$ to satisfy $(1-\epsilon)s_i \leq \widetilde{s}_i \leq (1+\epsilon)s_i$. Nevertheless, seeding this one-shot implementation with a discrete constant-factor solution computed from bicriteria centers and cell counts in $\widetilde{O}(n^{0.5}kd)$ time (Lemma H.13), we obtain an $\epsilon$-coreset of size $\widetilde{O}_p(k^{\frac{2p+2}{p+2}}\epsilon^{-2})$, consisting of input points only, in time $\widetilde{O}_p(\epsilon^{-1}n^{0.5}dk^{\frac{2p+3}{p+2}})$ (Theorem H.15). Compared to the previous work of Xue et al. [2023] where they obtain a coreset in $\widetilde{O}_p(n^{0.5}d^{1.5}k^{0.5}\epsilon^{-(p/2\vee 1)})$ time,

- Up to polylogarithmic factors, our algorithm is faster in the large dimension regime where $d \geq \Omega_p(k^{\frac{3p+4}{p+2}})$, i.e., $d \geq \Omega(k^{7/3})$ for $k$-median and $d \geq \Omega(k^{2.5})$ for $k$-means. The comparison is against their main bound, Theorem 3.1 of Xue et al. [2023]. In particular, the regime is independent of $\epsilon$.
- Our algorithm directly outputs a coreset of size $\widetilde{O}_p(k^{\frac{2p+2}{p+2}}\epsilon^{-2})$, while their main bound gives a coreset of size $\widetilde{O}_p(dk\epsilon^{-(2\vee p)})$. Obtaining our coreset size from their output by running the algorithm of Huang et al. [2024] on top would require an additional $\widetilde{O}_p(d^2\,\mathrm{poly}(k,\epsilon^{-p}))$ time.

As an application, we demonstrate that $(k,p)$-clustering can be used to bootstrap the construction of the data selection pipeline [Axiotis et al., 2024], as it enables the computation of approximate $k$-centers in sublinear time. Furthermore, we show that the quantum techniques developed for $(k,p)$-clustering can also be leveraged to obtain a sublinear-time quantum algorithm for data selection. We defer a more detailed discussion to Section H.

## 3 Conclusion

We present a quantum, sublinear-time algorithm for weighted sampling that yields a broad range of results in coreset construction. These include $(k,p)$-clustering, $\ell_p$ regression, $(k,p)$-subspace approximation, and low-rank approximation. For the low-rank approximation problem, we design specialized algorithms for multiple settings, including Frobenius norm error minimization, PSD low-rank approximation, kernel-based low-rank approximation, and tensor low-rank approximation. For $(k,p)$-clustering, we develop an improved quantum coreset construction that has a better $d$-dependence than the main bound of Xue et al. [2023] and outputs a coreset consisting of input points only (Theorem H.15), and we generalize this framework to address the data selection problem for foundation models. We highlight three major open problems arising from our work:

- **Two-sided approximation for clustering sensitivities.** Unlike regression and low-rank approximation, where coresets can be constructed efficiently via leverage scores or Lewis weights, the approximate sensitivities used in clustering are only known to be *upper bounds*. This asymmetry significantly limits the applicability of the recursive sampling framework to clustering. It remains an open question whether one can design algorithms that compute *two-sided* approximations to clustering sensitivities, unifying clustering with our weighted sampling framework.
- **Quantum algorithms for Frobenius norm tensor low-rank approximation.** While we achieve a $(1+\epsilon)$-approximation for matrix low-rank approximation in sublinear time, the scenario is more complex for tensors. For 3rd-order tensors, our bicriteria algorithm achieves a $(2+\epsilon)$-approximation with output rank $k^2$ (Theorem G.4), and our fixed-parameter algorithm achieves a $(2+\epsilon)$-approximation with exact output rank $k$ (Theorem G.7). A compelling open question is whether one can design a sublinear-time quantum algorithm that achieves a $(1+\epsilon)$-multiplicative approximation for tensor low-rank approximation. Remark G.5 gives an instance showing that the route of Theorem G.4, which fixes a $k$-dimensional span for the first factors in its first stage, cannot reach $(1+\epsilon)$.
- **Query lower bounds for coreset construction.** In Section I, we establish an $\Omega(\sqrt{nS})$ query lower bound for generic sensitivity sampling (Theorem I.1) that matches our framework up to $\mathrm{poly}(\epsilon^{-1},\log n)$ and VC-dimension factors. On the block family the $\epsilon$-dependence is settled at $\Theta(\sqrt{nS/\epsilon})$ up to a $\log S$ factor (Theorems I.3 and I.4), a factor $\epsilon^{-1/2}$ (up to logarithmic factors) below the framework's $\widetilde{O}(\epsilon^{-1}\sqrt{nS})$ on that family. An intriguing direction for future research is to capture the $\epsilon$-dependence when $S = o(\epsilon^{-2})$, since for larger $S$ the framework's $\epsilon^{-1}$ factor can be necessary (Section 2.3), and to extend these bounds to the remaining applications.

## Acknowledgments and Disclosure of Funding

David P. Woodruff would like to acknowledge support from a Simons Investigator Award and Office of Naval Research (ONR) award number N000142112647. Lichen Zhang is supported by a Simons Dissertation Fellowship in Mathematics. The authors declare no competing interests. Most of this paper was written by the authors and polished with the help of large language models (LLMs). Some of the results, including the lower bounds, the $\ell_p$ regression results, and the tighter bounds for $(k, p)$-clustering and $(k, p)$-subspace approximation, were improved with the help of LLMs (Claude Opus 5.5 in Claude Code) after the acceptance of the paper, and the authors verified all proofs.

## References

Deeksha Adil, Rasmus Kyng, Richard Peng, and Sushant Sachdeva. Fast algorithms for $\ell_p$-regression. *Journal of the ACM*, 71(5):34:1–34:45, 2024. doi: 10.1145/3686794.

Simon Apers and Ronald de Wolf. Quantum speedup for graph sparsification, cut approximation, and Laplacian solving. *SIAM Journal on Computing*, 51(6):1703–1742, 2022. doi: 10.1137/21M1391018.

Simon Apers and Sander Gribling. Quantum speedups for linear programming via interior point methods. *SIAM Journal on Computing*, 55(1):93–134, 2026. doi: 10.1137/25M1736098. Numbering as in arXiv:2311.03215v3.

Simon Apers, Sander Gribling, and Aaron Sidford. On computing approximate lewis weights. *arXiv preprint arXiv:2404.02881*, 2024.

Kyriakos Axiotis, Vincent Cohen-Addad, Monika Henzinger, Sammy Jerome, Vahab Mirrokni, David Saulpic, David P. Woodruff, and Michael Wunder. Data-efficient learning via clustering-based sensitivity sampling: Foundation models and beyond. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 2086–2107. PMLR, 2024.

Ainesh Bakshi, Nadiia Chepurko, and David P. Woodruff. Robust and sample optimal algorithms for PSD low rank approximation. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 506–516, 2020. doi: 10.1109/FOCS46700.2020.00054. Full version at arXiv:1912.04177.

Nikhil Bansal, Ola Svensson, and Luca Trevisan. New notions and constructions of sparsification for graphs and hypergraphs. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 910–928. IEEE, 2019.

Joshua Batson, Daniel A Spielman, and Nikhil Srivastava. Twice-Ramanujan sparsifiers. *SIAM Journal on Computing*, 41(6):1704–1721, 2012.

Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. doi: 10.1145/502090.502097.

András A. Benczúr and David R. Karger. Randomized approximation schemes for cuts and flows in capacitated graphs. *SIAM Journal on Computing*, 44(2):290–319, 2015. doi: 10.1137/070705970.

Mahdi Boroujeni, Soheil Ehsani, Mohammad Ghodsi, Mohammadtaghi Hajiaghayi, and Saeed Seddighin. Approximating edit distance in truly subquadratic time: Quantum and MapReduce. *Journal of the ACM*, 68(3):19:1–19:41, 2021. doi: 10.1145/3456807.

Jean Bourgain, Joram Lindenstrauss, and Vitali Milman. Approximation of zonoids by zonotopes. *Acta Mathematica*, 162(1–2):73–141, 1989. doi: 10.1007/BF02392835.

Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002. arXiv:quant-ph/0005055.

Vladimir Braverman, Dan Feldman, Harry Lang, Adiel Statman, and Samson Zhou. Efficient coreset constructions via sensitivity sampling. In Vineeth N. Balasubramanian and Ivor Tsang, editors, *Proceedings of The 13th Asian Conference on Machine Learning*, volume 157 of *Proceedings of Machine Learning Research*, pages 948–963. PMLR, 2021a.

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, 2021b.

Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theoretical Computer Science*, 312(1):3–15, 2004. doi: 10.1016/S0304-3975(03)00400-6.

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

Yanlin Chen, András Gilyén, and Ronald de Wolf. A Quantum Speed-Up for Approximating the Top Eigenvectors of a Matrix. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 994–1036, 2025. doi: 10.1137/1.9781611978322.29.

Nadiia Chepurko, Kenneth Clarkson, Lior Horesh, Honghao Lin, and David Woodruff. Quantum-inspired algorithms from randomized numerical linear algebra. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 3879–3900. PMLR, 17–23 Jul 2022.

Nai-Hui Chia, András Gilyén, Tongyang Li, Han-Hsuan Lin, Ewin Tang, and Chunhao Wang. Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning. *Journal of the ACM*, 69(5):33:1–33:72, 2022. doi: 10.1145/3549524.

Kenneth L Clarkson. Subgradient and sampling algorithms for $\ell_1$ regression. In *Symposium on Discrete Algorithms: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 257–266, 2005.

Kenneth L. Clarkson and David P. Woodruff. Low-rank approximation and regression in input sparsity time. *Journal of the ACM*, 63(6):54:1–54:45, 2017a. doi: 10.1145/3019134. Numbering as in arXiv:1207.6365v4.

Kenneth L. Clarkson and David P. Woodruff. Low-rank PSD approximation in input-sparsity time. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2061–2072, 2017b. doi: 10.1137/1.9781611974782.134.

Michael B. Cohen and Richard Peng. $\ell_p$ row sampling by lewis weights. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 183–192, 2015. doi: 10.1145/2746539.2746567. Full version: arXiv:1412.0588.

Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 163–172, New York, NY, USA, 2015a. Association for Computing Machinery. doi: 10.1145/2746539.2746569. Full version: arXiv:1410.6801.

Michael B Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 181–190, 2015b. Full version: arXiv:1408.5099.

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)*, pages 1758–1777. SIAM, 2017. Full version: arXiv:1511.07263v2.

Vincent Cohen-Addad, David Saulpic, and Chris Schwiegelshohn. A new coreset framework for clustering. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 169–182, New York, NY, USA, 2021. Association for Computing Machinery. doi: 10.1145/3406325.3451022. Full version: arXiv:2104.06133.

Vincent Cohen-Addad, Kasper Green Larsen, David Saulpic, and Chris Schwiegelshohn. Towards optimal lower bounds for k-median and k-means coresets. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1038–1051, 2022a. Full version: arXiv:2202.12793.

Vincent Cohen-Addad, Kasper Green Larsen, David Saulpic, Chris Schwiegelshohn, and Omar Ali Sheikh-Omar. Improved coresets for Euclidean $k$-means. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pages 2679–2694, 2022b. doi: 10.52202/068431-0194.

Anirban Dasgupta, Petros Drineas, Boulos Harb, Ravi Kumar, and Michael W Mahoney. Sampling algorithms and coresets for $\ell_p$ regression. *SIAM Journal on Computing*, 38(5):2060–2078, 2009. doi: 10.1137/070696507.

Vin de Silva and Lek-Heng Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008. doi: 10.1137/06066518X.

Petros Drineas, Michael W Mahoney, and S. Muthukrishnan. Sampling algorithms for $l_2$ regression and applications. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1127–1136, 2006. doi: 10.1145/1109557.1109682.

Christoph Dürr and Peter Høyer. A quantum algorithm for finding the minimum. *arXiv preprint quant-ph/9607014*, 1996.

Maryam Fazel, Yin Tat Lee, Swati Padmanabhan, and Aaron Sidford. Computing Lewis weights to high precision. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2723–2742. Society for Industrial and Applied Mathematics, 2022. doi: 10.1137/1.9781611977073.107. Numbering as in arXiv:2110.15563.

Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 569–578, New York, NY, USA, 2011. Association for Computing Machinery. doi: 10.1145/1993636.1993712. Numbering as in arXiv:1106.1379v4.

Shmuel Friedland and Anatoli Torokhti. Generalized rank-constrained matrix approximations. *SIAM Journal on Matrix Analysis and Applications*, 29(2):656–659, 2007.

Daniel Gibney, Ce Jin, Tomasz Kociumaka, and Sharma V. Thankachan. Near-optimal quantum algorithms for bounded edit distance and Lempel-Ziv factorization. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3302–3332. Society for Industrial and Applied Mathematics, 2024. doi: 10.1137/1.9781611977912.118.

András Gilyén, Zhao Song, and Ewin Tang. An improved quantum-inspired algorithm for linear regression. *Quantum*, 6:754, June 2022. ISSN 2521-327X. doi: 10.22331/q-2022-06-30-754. URL `https://doi.org/10.22331/q-2022-06-30-754`.

Sander Gribling, Aaron Sidford, and Chenyi Zhang. Computing Lewis weights to high precision using local relative smoothness. In *Proceedings of the Thirty Ninth Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pages 2901–2939. PMLR, 2026.

Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. Association for Computing Machinery. ISBN 0897917855. doi: 10.1145/237814.237866. URL `https://doi.org/10.1145/237814.237866`.

Lingxiao Huang and Nisheeth K. Vishnoi. Coresets for clustering in Euclidean spaces: importance sampling is nearly optimal. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pages 1416–1429, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369794. doi: 10.1145/3357713.3384296. Full version at arXiv:2004.06263.

Lingxiao Huang, Jian Li, and Xuan Wu. On optimal coreset construction for Euclidean $(k, z)$-clustering. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 1594–1604, New York, NY, USA, 2024. Association for Computing Machinery. doi: 10.1145/3618260.3649707. Full version at arXiv:2211.11923.

Arun Jambulapati, Yang P. Liu, and Aaron Sidford. Improved iteration complexities for overconstrained $p$-norm regression. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pages 529–542, 2022. doi: 10.1145/3519935.3519971.

Arun Jambulapati, James R Lee, Yang P Liu, and Aaron Sidford. Sparsifying sums of norms. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1953–1962. IEEE, 2023a.

Arun Jambulapati, Yang P Liu, and Aaron Sidford. Chaining, group leverage score overestimates, and fast spectral hypergraph sparsification. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 196–206, 2023b.

Arun Jambulapati, James R. Lee, Yang P. Liu, and Aaron Sidford. Sparsifying generalized linear models. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1665–1675, 2024. doi: 10.1145/3618260.3649684.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984. doi: 10.1090/conm/026/737400.

Michael Kapralov, Robert Krauthgamer, Jakab Tardos, and Yuichi Yoshida. Spectral hypergraph sparsifiers of nearly linear size. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1159–1170. IEEE, 2022.

Iordanis Kerenidis and Anupam Prakash. Quantum recommendation systems. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *LIPIcs*, pages 49:1–49:21. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017.

Iordanis Kerenidis, Jonas Landman, Alessandro Luongo, and Anupam Prakash. q-means: a quantum algorithm for unsupervised machine learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

Hartmut Klauck, Robert Špalek, and Ronald de Wolf. Quantum and classical strong direct product theorems and optimal time-space tradeoffs. *SIAM Journal on Computing*, 36(5):1472–1493, 2007. doi: 10.1137/05063235X.

Michael Langberg and Leonard J. Schulman. Universal $\epsilon$-approximators for integrals. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 598–607, 2010. doi: 10.1137/1.9781611973075.50.

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*, volume 23 of *Ergebnisse der Mathematik und ihrer Grenzgebiete (3)*. Springer-Verlag, Berlin, 1991.

James R Lee. Spectral hypergraph sparsification via chaining. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 207–218, 2023.

Troy Lee and Jérémie Roland. A strong direct product theorem for quantum query complexity. *Computational Complexity*, 22(2):429–462, 2013. doi: 10.1007/s00037-013-0066-8. Numbering as in arXiv:1104.4468v3.

Yin Tat Lee. *Faster algorithms for convex and combinatorial optimization*. PhD thesis, Massachusetts Institute of Technology, 2016.

Tongyang Li, Shouvanik Chakrabarti, and Xiaodi Wu. Sublinear quantum algorithms for training linear and kernel-based classifiers. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3815–3824. PMLR, June 2019. Numbering as in arXiv:1904.02276.

Chenghua Liu and Zhengfeng Ji. Accelerating regression tasks with quantum algorithms. In *Proceedings of the 43rd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2026. arXiv:2509.24757.

Tung Mai, Cameron Musco, and Anup Rao. Coresets for classification – simplified and strengthened. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 11643–11654. Curran Associates, Inc., 2021.

Ramgopal R. Mettu and C. Greg Plaxton. Optimal time bounds for approximate clustering. *Machine Learning*, 56(1–3):35–60, 2004. doi: 10.1023/B:MACH.0000033114.18632.e0.

Alexander Munteanu, Simon Omlor, and Christian Peters. p-generalized probit regression and scalable maximum likelihood estimation via sketching and coresets. In *International Conference on Artificial Intelligence and Statistics*, pages 2073–2100. PMLR, 2022.

Cameron Musco and Christopher Musco. Recursive sampling for the Nyström method. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 3833–3845. Curran Associates, Inc., 2017.

Cameron Musco and Kshiteej Sheth. Sublinear time low-rank approximation of Toeplitz matrices. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 5084–5117, 2024. doi: 10.1137/1.9781611977912.183.

Cameron Musco and David P. Woodruff. Sublinear time low-rank approximation of positive semidefinite matrices. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 672–683. IEEE, 2017. doi: 10.1109/FOCS.2017.68.

Cameron Musco, Christopher Musco, David P. Woodruff, and Taisuke Yasuda. Active linear regression for $\ell_p$ norms and beyond. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 744–753. IEEE, 2022. doi: 10.1109/FOCS54457.2022.00076.

Ashwin Nayak and Felix Wu. The quantum query complexity of approximating the median and related statistics. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC)*, pages 384–393, 1999. doi: 10.1145/301250.301349. Numbering as in arXiv:quant-ph/9804066v2.

Maris Ozols, Martin Roetteler, and Jérémie Roland. Quantum rejection sampling. *ACM Transactions on Computation Theory*, 5(3):11:1–11:33, 2013. doi: 10.1145/2493252.2493256.

Swati Padmanabhan, David P. Woodruff, and Qiuyi (Richard) Zhang. Computing approximate $\ell_p$ sensitivities. In *Advances in Neural Information Processing Systems*, volume 36, pages 36795–36825, 2023. doi: 10.52202/075280-1602.

Akbar Rafiey and Yuichi Yoshida. Sparsification of decomposable submodular functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10336–10344, 2022.

Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 143–152. IEEE, 2006. doi: 10.1109/FOCS.2006.37.

Gideon Schechtman and Artem Zvavitch. Embedding subspaces of $L_p$ into $l_p^n$, $0 < p < 1$. *Mathematische Nachrichten*, 227(1):133–142, 2001.

Poojan Chetan Shah and Ragesh Jaiswal. Quantum (inspired) $D^2$-sampling with applications. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zhao Song, David P Woodruff, and Peilin Zhong. Relative error tensor low rank approximation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2772–2789. SIAM, 2019. doi: 10.1137/1.9781611975482.172. Full version: arXiv:1704.08246v2.

Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

Daniel A Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011. doi: 10.1137/08074489X.

Michel Talagrand. Embedding subspaces of $L_p$ in $l_p^n$. In *Geometric Aspects of Functional Analysis: Israel Seminar (GAFA) 1992–94*, pages 311–326. Springer, 1995.

Kasturi Varadarajan and Xin Xiao. On the Sensitivity of Shape Fitting Problems. In Deepak D'Souza, Jaikumar Radhakrishnan, and Kavitha Telikepalli, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012)*, volume 18 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 486–497, Dagstuhl, Germany, 2012. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-939897-47-7.

David P. Woodruff and Taisuke Yasuda. Online lewis weight sampling. *ACM Transactions on Algorithms*, 21(4):39:1–39:50, 2025a. doi: 10.1145/3715127. Numbering as in arXiv:2207.08268v3.

David P. Woodruff and Taisuke Yasuda. Root ridge leverage score sampling for $\ell_p$ subspace approximation. In *Proceedings of the 66th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 2266–2291, 2025b. doi: 10.1109/FOCS63196.2025.00119. Full version: arXiv:2407.03262v3.

Yecheng Xue, Xiaoyu Chen, Tongyang Li, and Shaofeng H.-C. Jiang. Near-optimal quantum coreset construction algorithms for clustering. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 38881–38912. PMLR, 2023.

# A Additional Technical Overview

This section contains the parts of the technical overview of Section 2 deferred from the main text.

## A.1 Ridge Leverage Score and Lewis Weight Sampling

**Ridge leverage scores.** For $\ell_2$ regression the weights are the ordinary leverage scores, and instantiating the framework with them recovers the Grover-based sampler of Apers and Gribling [2026], which combines the recursive halving scheme of Cohen et al. [2015b] with a Johnson–Lindenstrauss sketch of the pseudoinverse [Johnson and Lindenstrauss, 1984, Spielman and Srivastava, 2011]. We do not elaborate on this known case. The instantiation that is new to this work is the *ridge leverage score* of Definition B.7, which governs low-rank approximation. The $k$-subspace sensitivity is at most $(1 + 1/k)$ times the ridge leverage score, as shown in Section C.3, and Cohen et al. [2017] showed that sampling by ridge leverage scores yields a projection-cost preserving sample. Compared with ordinary leverage scores, the new difficulty is that the regularization $\lambda_{A_k}$ is itself a data-dependent quantity, so a small reference set must simultaneously approximate the Gram matrix *and* the ridge term. Therefore, we define an $\epsilon$-approximator as an $\epsilon$-coreset of $A$ that also satisfies the additive-multiplicative spectral approximation guarantee of Section C.3. By Lemma C.21, such an approximator preserves every ridge leverage score up to a factor $1 \pm 4\epsilon$, which validates the oracle implemented from the child approximator in the recursion. Moreover, uniform halving followed by resampling against the weights with respect to the uniform half yields an $\epsilon$-approximator of size $O(\epsilon^{-2} k \log(k/\delta))$ by Lemma C.22, supplying conditions (C2) and (C3). The oracle itself is cheap: given a size-$s$ approximator $C$, Algorithm 6 stores two Gaussian sketches with $O(\log n)$ rows computed from the thin SVD of $C$, after which each query takes $\widetilde{O}(d)$ time. Plugging these ingredients into the framework yields an $\epsilon$-approximator of size $O(k\epsilon^{-2} \log k)$ in $\widetilde{O}(n^{0.5} d k^{0.5} \epsilon^{-1} + d k^{\omega-1})$ time by Corollary D.3, which we use for our low-rank approximation results in Section 2.4.

To adapt our sensitivity sampling framework for Lewis weights sampling, we encounter a notable challenge: given a coreset $B$ of $A$, it is guaranteed that any vector in the subspace of $A$ has its $\ell_p$ norm preserved by $B$, but the Lewis weights instead measure the $\ell_2$ norm of the subspace after a density transformation induced by $W^{1/2-1/p}$. Consequently, the Lewis weights of $A$ might not be preserved by $B$. We instead use the notion of an *$\epsilon$-approximator* of $A$ in Definition C.13, which preserves the Lewis quadratic form. Note that this is a different approximation notion than that of a coreset, as the cost becomes *global* rather than local: for generic sensitivity-based arguments, one relies on the fact that if $B$ is a coreset of $A$, then $B \cup \{a\}$ is also a coreset of $A \cup \{a\}$, but this is not true for an approximator of $A$, as adding a single row to both $A$ and $B$ would potentially affect the weights to all existing rows. In Cohen and Peng [2015], they provide a classical recursive sampling algorithm by noting that, if we sample according to the generalized Lewis weights with respect to an approximator, scaled up by $d^{p/2}$ when $p \geq 2$, then the resulting weighted sample is *also* an approximator, at constant accuracy (Lemma C.14). This is the approximator property our framework requires, and the argument of Cohen and Peng [2015] plays the role of the conditions (C2) and (C3). Theorem C.18 makes this formal. Thus, by using weighted sampling with Lewis weights, we generate a coreset for $\ell_p$ regression in $\widetilde{O}_p(\epsilon^{-1} n^{0.5} d^{1.5})$ time for $p < 2$ (as Theorem C.12 gives for $p = 2$) and in $\widetilde{O}_p(n^{0.5} d^{p/4+1}(\epsilon^{-1} + d^{0.5}))$ time for $p \geq 2$. For $p > 2$, this improves upon the prior quantum algorithm for Lewis weights sampling that is based on *iterating leverage scores* [Apers and Gribling,

2026], whose iteration is the classical scheme of Apers et al. [2024], Gribling et al. [2026]. For sampling, it suffices to approximate the Lewis weights up to a constant factor (Lemma C.17), which Theorem 4.2 of Apers and Gribling [2026] (arXiv v3) does after $\widetilde{O}_p(n^{0.5}d^7)$ preprocessing time for a dense matrix and $p \geq 2$. To construct a coreset, one then samples the $m = \widetilde{O}_p(d^{p/2}\epsilon^{-2})$ rows of Lemma C.17 with the Grover-based sampler of Lemma B.15, using $\widetilde{O}(\sqrt{nm})$ weight queries of $O(p^3d^{7/2})$ time each, so the prior runtime is $\widetilde{O}_p(n^{0.5}(d^7 + d^{p/4+3.5}\epsilon^{-1}))$. Our algorithm provides a speedup for any $p > 2$, as both terms of our runtime $\widetilde{O}_p(n^{0.5}d^{p/4+1}(\epsilon^{-1} + d^{0.5}))$ are at least a factor $d^2$ smaller than the second term of the prior runtime, and the additive $\widetilde{O}_p(d^{p/2+\omega})$ term of Theorem C.18 is lower order under $n \gg \mathrm{poly}(d)$. For $p < 2$ (which includes the popular $\ell_1$ regression), Theorem 4.2 of Apers and Gribling [2026] does not apply, and we compare with the concurrent work of Liu and Ji [2026] in the introduction and in Remark C.19. It is worth noting that the main purpose of the work of Apers and Gribling [2026] is to estimate Lewis weights up to $p = O(\log n)$ as they use it as a subroutine for solving linear programs, so their algorithm has no $p$ dependence on $d$, while our sampling algorithm is particularly suitable for small $p$.

### A.2 Recovering the Solution through the Subsampled Columns

Let $C \in \mathbb{R}^{d\times s}$ denote the weighted subset of columns of $A$ sampled by ridge leverage scores for $s = O(\epsilon^{-1}k\log k)$ (Lemma D.6). Then it is guaranteed that $\min_{X:\mathrm{rank}(X)\leq k} \|CX - A\|_F^2 \leq (1+\epsilon)\|A_k - A\|_F^2$. Constructing an optimal $X$ would require computing $P_k(C^\dagger A)$ where $P_k$ is the projection onto the top-$k$ principal components. Directly computing $C^\dagger A$ is too expensive, and standard approaches mostly involve using an *oblivious subspace embedding* (OSE) matrix, a random matrix that approximates the cost of all regression problems. Matrices such as CountSketch [Charikar et al., 2004, Clarkson and Woodruff, 2017a] could be applied in time $\mathrm{nnz}(A)$, but this is already too slow for our purpose. We address this with a pure-sampling framework for low-rank approximation: we demonstrate that it is possible to recover (or approximate) the solution $X$ via leverage score sampling.

In particular, let $S$ sample rows according to the leverage score distribution of $C$. A good rank-$k$ solution $CX^\star$ has column span inside that of $C$, so its row leverage scores are dominated by those of $C$ and $S$ oversamples them. The standard leverage score guarantee for unconstrained regression (Lemma B.13), applied to the design $CX^\star$, shows that $CX^\star(SCX^\star)^\dagger SA$ is a $(1+\epsilon)$-approximate rank-$k$ solution. Hence a good rank-$k$ solution exists in the row span of $SA$, and it is enough to solve the regression problem $\min_{Y:\mathrm{rank}(Y)\leq k} \|A - CYSA\|_F^2$. We further speed up the algorithm by employing two leverage score sampling matrices $T_1$ and $T_2$ on the left and right, respectively. For the new subsampled regression problem $\min_{Y:\mathrm{rank}(Y)\leq k} \|T_1AT_2 - T_1CYSAT_2\|_F^2$, we can compute the subsampled $A$ in *sublinear in* $n, d$ time, because $T_1AT_2$ and $SAT_2$ both amount to selecting a $\mathrm{poly}(k/\epsilon)$ subset of entries of $A$, which, assuming random access to the entries of $A$, can be done in the same order of time.

### A.3 Approximate Regression via Sampling Responses

For matrix low-rank approximation and its variants, ridge leverage score sampling is the crucial tool to compute a good approximate solution. Can we extend the framework to solve *tensor* low-rank approximation? Unfortunately, even for a 3rd order tensor $A \in \mathbb{R}^{n\times n\times n}$, it is not always the case that it admits a low-rank approximation, due to the so-called border rank issue [de Silva and Lim, 2008]. Even when the low-rank approximation exists, under the Exponential Time Hypothesis (ETH) no $2^{o(k^{1-o(1)})}$-time algorithm approximates the rank of $A$ within a constant factor, and no $2^{o(\epsilon^{-1/4})}$-time algorithm computes a $(1+\epsilon)$-approximate rank-1 approximation [Song et al., 2019, Theorems 1.6 and 1.7]. If one relaxes the problem by allowing the output to be a higher-rank solution (bicriteria solution) or a running time that depends exponentially on $k$ and $1/\epsilon$ (fixed-parameter tractable, i.e., FPT), then Song et al. [2019] provides algorithms with leading running time term being $\mathrm{nnz}(A)$. Their core algorithm is as follows: for tensor $A \in \mathbb{R}^{n\times n\times n}$, let $A_1, A_2, A_3 \in \mathbb{R}^{n\times n^2}$ be matrices such that the 1st, 2nd, and 3rd dimensions of the array are preserved, while the other 2 dimensions are collapsed and flattened into a dimension of size $n^2$. They then apply OSEs $S_1, S_2, S_3$ with only $\mathrm{poly}(k/\epsilon)$ columns to form $A_1S_1$, $A_2S_2$ and $A_3S_3$.

Although one might attempt to replace the OSEs $S_1, S_2$, and $S_3$ with leverage score matrices for $A_1, A_2$, and $A_3$, *this approach, unfortunately, does not work*. The argument of Song et al. [2019] is as follows: suppose the optimal rank-$k$ approximation $A_k$ exists, then $A_k = \sum_{i=1}^k U_i^* \otimes V_i^* \otimes W_i^*$. To reduce the problem dimension, the goal is to demonstrate that a good approximate solution exists in the column span of $A_1 S_1$ and $A_2 S_2$. In particular, suppose we have access to $V^*$ and $W^*$, and let $Z_1 \in \mathbb{R}^{k \times n^2}$ have $V_i^* \otimes W_i^*$ as its $i$-th row. Then the optimal $U^*$ could be recovered by solving $\min_{U \in \mathbb{R}^{n \times k}} \|UZ_1 - A_1\|_F^2$, as $\|UZ_1 - A_1\|_F^2 = \|\sum_{i=1}^k U_i \otimes V_i^* \otimes W_i^* - A\|_F^2$. This multiple response regression problem can be accelerated by applying an OSE on the right and instead solving $\min_U \|UZ_1 S_1 - A_1 S_1\|_F^2$, and its optimal solution $\widehat{U} = A_1 S_1 (Z_1 S_1)^\dagger$ is in the column span of $A_1 S_1$.

For sampling, this setup is more challenging. If we were to replace $S_1$ with the leverage score sampling matrix, we would require the *leverage score matrix of* $Z_1$ in order to preserve the cost of the optimal solution, and it is unclear how to sample according to the leverage scores of the unknown matrix $Z_1$. On the other hand, we do have access to the response matrix $A_1$. However, a simple counterexample demonstrates that sampling directly from $A_1$ fails: suppose $A_1$ is a single column equal to $e_n$, and the design matrix $Z_1$ is $e_i + e_n$ for $i$ randomly chosen from 1 to $n-1$. Any sampling scheme based on $A_1$ will likely sample the $n$-th entry but miss the $i$-th entry with high probability. This would lead to a solution to the original problem that has twice the optimal cost.

Surprisingly, we show that this factor-2 loss is essentially the worst that can happen: if one instead samples from the ridge leverage score distribution of $A_1$, then there exists a solution $\widehat{U}$ in the column span of $A_1 S_1$ ($S_1$ is the ridge leverage score sampling matrix of $A_1$) such that $\|\widehat{U} Z_1 - A_1\|_F^2 \leq (2+\epsilon) \cdot \min_U \|UZ_1 - A_1\|_F^2$. This result is particularly surprising as one might expect an adversarial choice of $A_1$ that would disrupt ridge leverage score sampling. However, ridge leverage scores provide the so-called projection-cost preserving guarantee: for any rank-$k$ projection $P$, we have that $(1-\epsilon)\|(I-P)A_1\|_F^2 \leq \|(I-P)A_1 S_1\|_F^2 \leq (1+\epsilon)\|(I-P)A_1\|_F^2$, where setting $P_k$ as the projection onto the top-$k$ principal components of $A_1 S_1$ minimizes $\|(I-P_k)A_1 S_1\|_F^2$. Additionally, the optimal cost of the regression can be bounded by $\|[A_1]_k - A_1\|_F^2$. Setting $\widehat{U} = P_k A_1 Z_1^\dagger$, we get

$$\begin{aligned}
\|\widehat{U} Z_1 - A_1\|_F^2 =& \|P_k A_1 Z_1^\dagger Z_1 - A_1\|_F^2 \\
=& \|(P_k A_1 - A_1)(Z_1^\dagger Z_1) - A_1(I - Z_1^\dagger Z_1)\|_F^2 \\
=& \|(P_k A_1 - A_1)(Z_1^\dagger Z_1)\|_F^2 + \|A_1(I - Z_1^\dagger Z_1)\|_F^2 \\
\leq& \|(I-P_k)A_1\|_F^2 + \|A_1(I - Z_1^\dagger Z_1)\|_F^2 \\
\leq& (2+\epsilon)\,\mathrm{OPT},
\end{aligned}$$

where $\mathrm{OPT} := \min_U \|UZ_1 - A_1\|_F^2$, and we use the Pythagorean theorem in the proof, along with the fact that $\|A_1 - A_1 Z_1^\dagger Z_1\|_F^2$ is the optimal cost. To see $\widehat{U}$ is in the column span of $A_1 S_1$, it is enough to observe that $P_k$ is the projection onto the top-$k$ principal components of $A_1 S_1$, and hence $\widehat{U}$ is in the column span of $P_k$, a subset of the column span of $A_1 S_1$. Our bicriteria result pays the response-sampling loss only once: after the first stage fixes a $k$-dimensional span for the first factors, we treat the second and third modes *jointly* as a single matrix low-rank approximation problem on an $n \times kn$ matrix. This gives a $(2+\epsilon)$ ratio with output rank $k^2$ by Theorem G.4, and the same first stage followed by the classical $(1+\epsilon)$ fixed-parameter algorithm of Song et al. [2019] gives exact output rank $k$ with ratio $(2+\epsilon)$ by Theorem G.7, whose hypotheses include that an optimal rank-$k$ solution exists.

# B Preliminaries

## B.1 Notation

For any $n \in \mathbb{N}$, let $[n]$ denote the set $\{1, 2, \ldots, n\}$. We use $\widetilde{O}(\cdot)$ to hide polylogarithmic factors in $n$, $d$, $1/\epsilon$, $1/\delta$, and other problem-related parameters, such as $k$ and $p$. We use $O_p(\cdot)$ and $\widetilde{O}_p(\cdot)$ when the hidden constant may also depend on $p$, and in our clustering bounds this dependence is exponential in $p$. Throughout the paper, we assume that $n$ and $d$ are polynomially larger than $k$ and $1/\epsilon$, and we write $n, d \gg \mathrm{poly}(k, 1/\epsilon)$. For the regression results, we also assume that $n$ is polynomially larger

than $d$. The theorems state their full bounds, while the abstract, introduction, and tables report the leading term under these assumptions. For instance, the first term of Theorem D.8 is the leading term once $d \geq \epsilon^{-3.5}k$. For two numbers $a$ and $b$, we use $a \vee b$ as a shorthand for $\max\{a, b\}$. We use $a = (1 \pm \epsilon)b$ to denote $a \in [(1-\epsilon)b, (1+\epsilon)b]$. We use $\omega$ to denote the matrix multiplication exponent.

For a matrix or a tensor $A$, we use $\|A\|_2$ or simply $\|A\|$ to denote the spectral norm of $A$, which for a tensor is

$$\|A\| = \sup_{x,y,z \neq 0} \frac{|A(x,y,z)|}{\|x\| \cdot \|y\| \cdot \|z\|}.$$

Let $A \in \mathbb{R}^{n \times d}$ and $k \leq \min\{n, d\}$. We will use $A_k$ or $[A]_k$ to denote its best rank-$k$ approximation. Let $\|A\|_F$ denote the Frobenius norm of a matrix/tensor $A$, the square root of the sum of squares of its entries. For $1 \leq p < 2$, we use $\|A\|_p$ to denote the entry-wise $\ell_p$-norm of a matrix/tensor $A$, the $p$-th root of the sum of $p$-th powers of the absolute values of its entries.

Let $\text{nnz}(A)$ denote the number of nonzero entries of $A$, $A^\top$ its transpose and $A^\dagger$ its Moore-Penrose pseudoinverse. For a square matrix $A$, let $\det(A)$ denote its determinant, and let $A^{-1}$ denote its inverse when $A$ has full rank.

For a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, we use $A_{i,j,l}$ to denote its $(i,j,l)$-th element. We call the three kinds of fibers $A_{*,j,l}$ (mode-1), $A_{i,*,l}$ (mode-2), and $A_{i,j,*}$ (mode-3) its columns, rows, and tubes, respectively. Note that the columns of the unfolding $A_1$ are the columns of $A$ in this sense.

A tensor $A$ is symmetric if and only if for any $i, j, k$, $A_{i,j,k} = A_{i,k,j} = A_{j,i,k} = A_{j,k,i} = A_{k,i,j}$.

For a tensor $A \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we use $\top$ to denote rotation (3-dimensional transpose) so that $A^\top \in \mathbb{R}^{n_3 \times n_1 \times n_2}$. For a tensor $A \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and matrix $B \in \mathbb{R}^{n_3 \times k}$, we define the tensor-matrix dot product to be $A \cdot B \in \mathbb{R}^{n_1 \times n_2 \times k}$.

### B.2 Sensitivity and Coreset

Throughout this paper, we will extensively work with sensitivities and coresets. Let $X$ be some universe of elements. Our main focus is the cost function: $\text{cost} : \mathbb{R}^d \times X \to \mathbb{R}_{\geq 0}$, which measures the cost of an element $x \in X$ with respect to the first argument. We then define the notion of strong and weak coresets.

**Definition B.1** ((Strong) Coreset)**.** *Let $B \subseteq A$ and $\epsilon \in (0,1)$. We say that $B$ is an* $\epsilon$-strong coreset *or $\epsilon$-coreset of $A$ if there exists a nonnegative weight vector $w \in \mathbb{R}_{\geq 0}^{|B|}$ such that for all $x \in X$,*

$$\sum_{b \in B} w_b \cdot \text{cost}(b, x) = (1 \pm \epsilon) \cdot \text{cost}(A, x).$$

A strong coreset preserves the cost over all possible $x \in X$, but sometimes we only need the optimal cost preserved. We also introduce the notion of weak coreset.

**Definition B.2** (Weak Coreset)**.** *Let $B \subseteq A$ and $\epsilon \in (0,1)$. We say that $B$ is an* $\epsilon$-weak coreset *if there exists a nonnegative weight vector $w \in \mathbb{R}_{\geq 0}^{|B|}$ such that*

$$\min_{x \in X} \sum_{b \in B} w_b \cdot \text{cost}(b, x) = (1 \pm \epsilon) \cdot \text{OPT},$$

*where* $\text{OPT} = \min_{x \in X} \text{cost}(A, x)$.

**Remark B.3.** *Oftentimes, given a weighted subset $(B, w)$, we will use $\text{cost}(B, x)$ as an abbreviation for $\sum_{b \in B} w_b \cdot \text{cost}(b, x)$, as our analysis and algorithms on the subset of points work in both unweighted and weighted settings. Hence, when the weight is clear from context, we will abuse notation and use $\text{cost}(b, x)$ to denote $w_b \cdot \text{cost}(b, x)$.*

**Definition B.4** (Sensitivity and Generalized Sensitivity)**.** *Let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$. We define the* sensitivity of $a_i$ *as*

$$s_i(A, A) = \max_{x \in X} \frac{\text{cost}(a_i, x)}{\text{cost}(A, x)}.$$

*Let $B \subset \mathbb{R}^d$. We define the* sensitivity of $a_i$ with respect to $B$ *as*

$$s_i(A, B) = \max_{x \in X, \mathrm{cost}(B,x) \neq 0} \frac{\mathrm{cost}(a_i, x)}{\mathrm{cost}(B, x)}.$$

### B.3 Leverage Score, Ridge Leverage Score, and Lewis Weights

**Definition B.5** (Statistical Dimension)**.** *For real value $\lambda \geq 0$ and a rank-$d$ matrix $A \in \mathbb{R}^{n \times d}$ with singular values $\sigma_i(A)$, the quantity $s_d^\lambda(A) := \sum_{i=1}^d \frac{1}{1+\lambda/\sigma_i^2(A)}$ is the statistical dimension of the ridge regression problem with regularizing weight $\lambda$.*

**Definition B.6** (Leverage Score)**.** *Given matrix $A \in \mathbb{R}^{n \times d}$, leverage score can be defined as follows:*

$$\tau_i(A) := a_i^\top (A^\top A)^\dagger a_i,$$

*where $a_i^\top$ is the $i$-th row of $A$ for all $i \in [n]$.*

**Definition B.7** (Ridge Leverage Score)**.** *Given matrix $A \in \mathbb{R}^{n \times d}$, we denote the $i$-th ridge leverage score, for $i \in [n]$, as follows:*

$$\overline{\tau}_i(A, \lambda_{A_k}) := a_i^\top (A^\top A + \lambda_{A_k} I)^\dagger a_i,$$

*where $\lambda_{A_k} = \|A - A_k\|_F^2 / k$ and $I \in \mathbb{R}^{d \times d}$ is the identity matrix (the pseudo-inverse coincides with the inverse unless $\mathrm{rank}(A) \leq k$, in which case $\lambda_{A_k} = 0$ and the score is the leverage score). When the rank $k$ is clear from context, we may abbreviate $\overline{\tau}_i(A)$ as $\overline{\tau}_i(A, \lambda_{A_k})$.*

**Definition B.8** (Generalized Ridge Leverage Score)**.** *Let $A \in \mathbb{R}^{n \times d}$, $C \in \mathbb{R}^{n \times d'}$, and $i \in [d]$. We define the $i$-th generalized ridge leverage score of $A \in \mathbb{R}^{n \times d}$ with respect to $C \in \mathbb{R}^{n \times d'}$ as follows:*

$$\overline{\tau}_i(A, C, \lambda_{C_k}) = \begin{cases} a_i^\top (CC^\top + \lambda_{C_k} I_n)^\dagger a_i, & \text{if } a_i \in \mathrm{span}(CC^\top + \lambda_{C_k} I_n), \\ \infty, & \text{otherwise,} \end{cases}$$

*where $a_i \in \mathbb{R}^n$ denotes the $i$-th column of $A$ and $\lambda_{C_k} = \|C - C_k\|_F^2 / k$ as in Definition B.7. When the rank $k$ is clear from context, we may use $\overline{\tau}_i(A, C)$ as shorthand for $\overline{\tau}_i(A, C, \lambda_{C_k})$.*

**Definition B.9** (Lewis Weights)**.** *Let $p \in (0, \infty)$ and $A \in \mathbb{R}^{n \times d}$. We define the $\ell_p$ Lewis weights of $A$, denoted by $w_A$, as*

$$w_{A,i} = \tau_i(W_A^{1/2 - 1/p} A),$$

*or equivalently,*

$$w_{A,i}^{2/p} = a_i^\top (A^\top W_A^{1-2/p} A)^{-1} a_i.$$

### B.4 Matrix Approximations

**Definition B.10** (Subspace Embedding in Sarlós [2006])**.** *Let $\epsilon$, $\delta \in (0, 1)$ and $n > d$. Given a matrix $U \in \mathbb{R}^{n \times d}$ which is orthonormal (i.e., $U^\top U = I_d$), we say $S \in \mathbb{R}^{m \times n}$ is an $\mathsf{SE}(\epsilon, \delta, n, d)$ subspace embedding for fixed $U$ if, for all $x \in \mathbb{R}^d$,*

$$(1 - \epsilon)\|Ux\|_2^2 \leq \|SUx\|_2^2 \leq (1 + \epsilon)\|Ux\|_2^2$$

*holds with probability $1 - \delta$. This is equivalent to*

$$\|U^\top S^\top S U - U^\top U\| \leq \epsilon.$$

**Definition B.11** (Weak $\epsilon$-Affine Embedding, Theorem 39 in Clarkson and Woodruff [2017a])**.** *Let matrices $A \in \mathbb{R}^{n \times r}$ and $B \in \mathbb{R}^{n \times d}$. Given matrix $S \in \mathbb{R}^{t \times n}$, we say $S$ is a weak $\epsilon$-affine embedding if the following holds: let $\widehat{X} = \arg\min_X \|AX - B\|_F^2$ and $\widehat{B} = A\widehat{X} - B$. Then, for all $X \in \mathbb{R}^{r \times d}$,*

$$\|S(AX - B)\|_F^2 - \|S\widehat{B}\|_F^2 = (1 \pm \epsilon)\|AX - B\|_F^2 - \|\widehat{B}\|_F^2.$$

### B.5 Properties of Leverage Score

Sampling according to leverage score distribution yields a weak affine embedding property. Additionally, solving the subsampled problem results in an optimal solution whose cost is close to the original optimal cost.

**Lemma B.12** (Theorem 42 in Clarkson and Woodruff [2017a])**.** *Let matrix $A \in \mathbb{R}^{n\times r}$ with rank at most $k$, and let $B \in \mathbb{R}^{n\times d}$. If $S \in \mathbb{R}^{n\times n}$ is a sampling and rescaling diagonal matrix according to the leverage scores of $A$, let $m = O(\epsilon^{-2}k\log k)$ denote the number of nonzero entries on the diagonal of $S$. Then, with probability at least $0.99$, for all $X \in \mathbb{R}^{r\times d}$, we have:*

- *$S$ is a weak $\epsilon$-affine embedding (see Definition B.11).*
- *Equivalently, if $\widehat{X} = \arg\min_X \|AX - B\|_F^2$, $\widehat{B} = A\widehat{X} - B$, and $C := \|S\widehat{B}\|_F^2 - \|\widehat{B}\|_F^2$, then*

$$(1-\epsilon)\cdot\|AX - B\|_F^2 + C \le \|S(AX - B)\|_F^2 \le (1+\epsilon)\cdot\|AX - B\|_F^2 + C.$$

**Lemma B.13** (Leverage Score Preserves Optimal Cost, Lemma C.31 of Song et al. [2019])**.** *Let $A \in \mathbb{R}^{n\times r}$ be a matrix with rank at most $k$, and let $B \in \mathbb{R}^{n\times d}$. If we sample $O(k\log k + k/\epsilon)$ rows of $A$ and $B$ proportional to the leverage scores of $A$ to obtain a sampling matrix $S$, then with probability at least $0.99$,*

$$\|AY_* - B\|_F^2 \le (1+\epsilon)\cdot\min_Y \|AY - B\|_F^2,$$

*where $Y_* = \arg\min_Y \|SAY - SB\|_F^2$.*

**Remark B.14.** *Lemmas B.12 and B.13 are proved in Song et al. [2019] for a constant failure probability, with $0.99$ in their Theorem C.50 and $0.999$ in their Lemma C.31. We read the $\delta$-dependence off their proofs, which combine two events. The first event is a subspace embedding by a matrix Chernoff bound, see their Lemma C.22, and the second event is an approximate matrix product for the residual by a second-moment bound and Markov's inequality, see their Lemma C.29. Hence, a failure probability $\delta$ enters only the sample size, as $\log(1/\delta)$ through the first event and as $1/\delta$ through the second event. Every invocation in this paper that runs either lemma with a failure probability $\delta$ uses a specific constant $\delta$, such as $0.04$, $0.004$, $0.002$, or $0.001$. This changes the sample size by a constant factor, and this factor is absorbed by the $O(\cdot)$.*

### B.6 Quantum Primitives

Our core quantum primitive is a sampling algorithm based on Grover search.

**Lemma B.15** (Claim 3 in Apers and de Wolf [2022])**.** *Let $n$ be a positive integer, let $q_1, \ldots, q_n \in [0,1]$, each specified with $O(\log n)$ bits of precision, and let $\delta \in (0,1)$. There is a quantum algorithm, $\textsc{QSample}(q,\delta)$, that with probability at least $1-\delta$ outputs a set $S \subseteq [n]$ containing each index $i$ independently with probability $q_i$ (precisely: its output coincides, except on an event of probability at most $\delta$, with such a sample), and runs in expected time $\widetilde{O}(\sqrt{n\max\{1,\sum_{i=1}^n q_i\}}\log(1/\delta))\cdot\mathcal{T}$ over its internal randomness, where $\mathcal{T}$ is the time to compute $q_i$. The precision hypothesis is without loss of generality throughout this paper: rounding each $q_i$ up to a multiple of $n^{-2}$ preserves the overestimate property of our sampling probabilities and increases $\sum_{i=1}^n q_i$ by at most $1/n$.*

We note that this runtime bound could also be achieved via quantum rejection sampling [Ozols et al., 2013]. Let $P = \sum_{i=1}^n q_i$, then $q_i/P$ for all $i \in [n]$ induces a probability distribution, which we denote by $\sigma$. Recall that rejection sampling aims to generate one sample from the target distribution $\sigma$ (where $\sigma_i = q_i/P$) using a uniform proposal distribution $\pi$ (where $\pi_i = 1/n$). The query complexity is $\widetilde{O}(\max_{i\in[n]}\sqrt{\sigma_i/\pi_i})\cdot\mathcal{T}$. Since each $q_i \le 1$, the ratio can be upper bounded by $\max_i(q_i/P)/(1/n) \le n/P$. Thus, the complexity to generate *one sample* is $\widetilde{O}(\sqrt{n/P})\cdot\mathcal{T}$. As $\sum_{i=1}^n q_i = P$, if we choose each index $i$ with probability $q_i$ independently, then the expected size is $P$, hence the total expected complexity is $\widetilde{O}(P\sqrt{n/P})\cdot\mathcal{T} = \widetilde{O}(\sqrt{nP})\cdot\mathcal{T}$, as desired. Claim 3 of Apers and de Wolf [2022] is stated in expected time without a failure probability, but it relies on the bounded-error search of their Claim 2, so we obtain the success probability $1-\delta$ of Lemma B.15 as follows. The number $K$ of marked indices is a sum of independent Bernoulli$(q_i)$ variables with mean $P$. For $j = 1, 2, \ldots$, the $j$-th Grover search is the search of Theorem 3 of Brassard et al.

[2002] for a marked index among those not yet found, run with the cutoff $c\sqrt{n}\log(4j^2/\delta)$ for a suitable constant $c$, and the algorithm stops at the first search that reaches its cutoff without finding a marked index. When $t \geq 1$ marked indices remain, the current search finds one in expected time $C\sqrt{n/t}\cdot\mathcal{T}$ for an absolute constant $C$, and we run it in independent windows of $2C\sqrt{n}\cdot\mathcal{T}$ steps, restarting after each window. By Markov's inequality, a window succeeds with probability at least $1/2$, as $t \geq 1$, so with $c := 2C/\log 2$, the cutoff contains $\log_2(4j^2/\delta)$ windows and is reached with probability at most $\delta/(4j^2)$. Moreover, the expected time is still $O(\sqrt{n/t})\cdot\mathcal{T}$, since, by Markov's inequality, the first window is left with probability at most $1/(2\sqrt{t})$, and the restarts then cost at most $2\cdot 2C\sqrt{n}\cdot\mathcal{T}$ in expectation. When no marked index remains, the search reaches the cutoff and the algorithm stops correctly. The total failure probability is at most $\sum_{j\geq 1}\delta/(4j^2) < \delta/2$. Conditioned on $K$, the expected running time is $\sum_{j=1}^{K} O(\sqrt{n/(K-j+1)})\cdot\mathcal{T} + O(\sqrt{n}\log(4(K+1)^2/\delta))\cdot\mathcal{T} = O(\sqrt{nK}+\sqrt{n}\,(\log(K+1)+\log(1/\delta)))\cdot\mathcal{T}$. Since $\mathbb{E}\sqrt{K} \leq \sqrt{P}$ and $K \leq n$, this is $\widetilde{O}(\sqrt{n\max\{1,P\}}\log(1/\delta))\cdot\mathcal{T}$. Since a candidate is checked with one query, a failed search can only miss marked indices and never reports an unmarked one. Hence, the output is a subset of the exact sample and coincides with it on the success event.

*Convention.* Running-time and query bounds for algorithms that call QSAMPLE are expected values over the internal randomness of the sampler calls. Theorems C.3 and C.12 state this explicitly, with the expectation also taken over the random halves of the recursion and conditioned on the success event. A theorem that states a running time without the word "expected" should be read in this sense. Each proof below that invokes QSAMPLE splits an explicit share of its failure budget equally among these calls, and the factor $O(\log(K/\delta))$ in the running time of $K$ calls is absorbed by $\widetilde{O}(\cdot)$. Since the output of a failed call is an arbitrary subset, every event on which sizes are bounded in expectation includes the success of the sampler calls. For an expectation conditioned on a success event $\mathcal{F}$ with $\Pr[\mathcal{F}] \geq 1/2$ that may depend on the outcome of a call, we use $\mathbb{E}[X \mid \mathcal{F}] \leq \mathbb{E}[X\,\mathbf{1}_{\mathcal{F}'}]/\Pr[\mathcal{F}] \leq 2\,\mathbb{E}[X\,\mathbf{1}_{\mathcal{F}'}]$, where $\mathcal{F}' \supseteq \mathcal{F}$ is an event determined before the call, so the expected running time of the call given its input applies. In the algorithm boxes, we write QSAMPLE$(q)$ when the failure parameter is fixed in the accompanying proof.

Throughout the paper, we will use the notation QLS$(A, s, \delta)$ to denote Algorithm 4, the procedure of Theorem C.12 that samples rows or columns of $A$ according to the leverage scores with size parameter $s$. Its guarantees and the cost of computing the scores are those of Theorem C.12, where $d$ is the number of columns (resp. rows) of $A$, and the running time is expected in the sense of Theorem C.3. The sampling step takes $\widetilde{O}(\sqrt{ns})\cdot\mathcal{T}$ time, where $\mathcal{T}$ is the time to compute a single score. Similarly, we use QRLS$(A, k, \epsilon, \delta)$ to denote the procedure of Theorem C.25 that samples according to the rank-$k$ ridge leverage scores, and QRLS$_s(A, k, s, \delta)$ to denote its fixed-size variant in Corollary D.3.

# C A Quantum Recursive Sampling Framework for Coreset

Throughout this section, let us consider $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$ to be a set of $n$ points in $\mathbb{R}^d$, and $X$ to be a set. Let $\mathrm{cost} : \mathbb{R}^d \times X \to \mathbb{R}_{\geq 0}$ be a cost function, and for $x \in X$, let $\mathrm{cost}(A, x) = \sum_{i=1}^{n}\mathrm{cost}(a_i, x)$. The main objective of this section is to develop a framework for sampling a weighted subset of $A$ that approximates the cost of $A$. To do so, we prove that if the weights satisfy certain assumptions, then a generic recursive sampling framework could construct a coreset from these weights. The assumptions are listed in the following.

We use two pieces of notation throughout this section. First, for a finite set $A$, we write $A' \subset_{1/2} A$ for the random subset of $A$ that contains each point of $A$ independently with probability $1/2$. Applying this operation $T$ times gives a chain $A_0 \subset_{1/2} A_1 \subset_{1/2} \cdots \subset_{1/2} A_T = A$. Thus, a point of $A$ belongs to $A_t$ with probability $2^{-(T-t)}$, and $|A_t|$ is a sum of independent indicators with mean $n2^{-(T-t)}$. Equivalently, every $a_i \in A$ is assigned $T$ independent unbiased random bits, and $a_i \in A_t$ if and only if its first $T-t$ bits all equal one. Following Apers and Gribling [2026], we realize the chain in this way, and the random bits are simulated as described in the quantum computation model in the introduction. Second, given $A = \{a_1, \ldots, a_n\}$, a subset $A' \subseteq A$, and a weighted finite set $B$, we define the *augmented weights*

$$w_i'(A, A'; B) := \begin{cases} w_i(A, B), & \text{if } a_i \in A', \\ w_i(A, B \cup \{a_i\}), & \text{otherwise,} \end{cases}$$

where $a_i$ joins $B$ with weight one, and we abbreviate $w_i'(A, A') := w_i'(A, A'; A')$. In the algorithms below, $A'$ is the uniform half of $A$ and $B$ is a coreset of $A'$, and we read off the applicable case from the membership bit of $a_i$.

**Assumption C.1.** *Given two finite subsets $A, B \subseteq \mathbb{R}^d$, let $w(A, B) \in \mathbb{R}^{|A|}$ be a nonnegative weight vector where $w_i(A, B)$ is the weight of $a_i$ with respect to $B$, and we assume that $w_i(A, B)$ depends on $A$ only through the point $a_i$, so that $w_i(A, B) = w_j(A', B)$ whenever $a_i$ is the $j$-th point of a finite set $A' \subseteq \mathbb{R}^d$, and in particular $w_i(A, A_S) = w_i(A_S, A_S)$ for $i \in S$. We assume $w$ satisfies the following conditions for every finite point set $A \subseteq \mathbb{R}^d$, with $n = |A|$ and $[n]$ indexing its points, where $B$, and $C$ in item 4, may carry weights:*

- *Consistent total weights: for any subset $S \subseteq [n]$, $\sum_{i \in S} w_i(A_S, A_S) \leq \mathrm{sum}(w)$ where $\mathrm{sum}(w)$ is a finite upper bound on the total weights.*
- *Uniform sampling bound: for every subset $A' \subseteq A$, the augmented weights satisfy $w_i'(A, A') = w_i'(A, A'; A') \geq w_i(A, A)$ for all $i \in [n]$.*
- *Importance sampling bound: for $\epsilon, \delta \in (0, 1)$ there is a quantity $g(\epsilon, \delta, n, d)$ with the following property. Let $u \in \mathbb{R}^n$ satisfy $u_i \geq w_i(A, A)$ for all $i \in [n]$, and let $B$ be the weighted subset of $A$ obtained by including each $a_i$ independently with probability $q_i = \min\{1, g(\epsilon, \delta, n, d) \cdot u_i\}$ and reweighting each included point by $1/q_i$. Then $B$ has expected size at most $g(\epsilon, \delta, n, d) \cdot \|u\|_1$, and with probability at least $1 - \delta$, $B$ is an $\epsilon$-coreset of $A$.*
- *Coreset preserves weights: let $B$ be an $\epsilon$-coreset of $A$, then $w_i(C, B) \in [(1+\epsilon)^{-1}, (1-\epsilon)^{-1}] \cdot w_i(C, A)$ for any fixed finite $C \subseteq \mathbb{R}^d$ and for every point $c_i$ of $C$.*

We make two remarks on the importance sampling bound. First, for any $\alpha \geq g := g(\epsilon, \delta, n, d)$, the vector $(\alpha/g) \cdot u$ is again an entrywise overestimate of $w(A, A)$, so the bound applies verbatim to the probabilities $q_i = \min\{1, \alpha \cdot u_i\}$, with expected size at most $\alpha \cdot \|u\|_1$. The algorithms below use $\alpha = c \cdot g$ for a constant $c$, and we may assume $g \cdot \mathrm{sum}(w) \geq 1$ by increasing $g$. Second, we take $g$ nondecreasing in $n$, hence $g(\epsilon, \delta, n, d)$ may be used for every subset of $A$ as well. In our instantiations, $g$ depends on $\delta$ only through $\log(1/\delta)$, as in Lemma C.6.

**Algorithm 2** Quantum iterative sampling for coreset.

**procedure** QITERATESAMPLE($A \in \mathbb{R}^{n \times d}, \epsilon, \delta$)
  $c \leftarrow 1000$
  $\epsilon' \leftarrow 0.01, \delta' \leftarrow \delta/(2\lceil \log_2 n \rceil)$ ▷ $\delta' \leq \delta/(2T)$ for the $T$ below
  $s \leftarrow 4c \cdot g(\epsilon, \delta/2, n, d) \cdot \mathrm{sum}(w)$
  $s' \leftarrow 4c \cdot g(\epsilon', \delta', n, d) \cdot \mathrm{sum}(w)$
  $T \leftarrow \max\{1, \lceil \log_2(n/s') \rceil\}$
  $A_0 \subset_{1/2} A_1 \subset_{1/2} \dots \subset_{1/2} A_{T-1} \subset_{1/2} A_T = A$
  $C_0 \leftarrow A_0$ ▷ learned explicitly by QSAMPLE($\mathbf{1}[a_i \in A_0], \delta'/(T+1)$)
  **for** $t = 1 \to T - 1$ **do**
    Implement a classical oracle for $w_i'(A_t, A_{t-1}; C_{t-1})$ for all $a_i \in A_t$
      ▷ $q_i = \min\{1, c \cdot g(\epsilon', \delta', n, d) \cdot w_i'(A_t, A_{t-1}; C_{t-1})\}$
    $D_t \leftarrow$ QSAMPLE($q, \delta'/(T+1)$)
    $C_t \leftarrow D_t^\top A_t$ ▷ $C_t \in \mathbb{R}^{\|D_t\|_0 \times d}$
  **end for**
  Implement a classical oracle for $w_i'(A_T, A_{T-1}; C_{T-1})$ for all $a_i \in A_T$
    ▷ $q_i = \min\{1, c \cdot g(\epsilon, \delta/2, n, d) \cdot w_i'(A_T, A_{T-1}; C_{T-1})\}$
  $D_T \leftarrow$ QSAMPLE($q, \delta'/(T+1)$)
  $C_T \leftarrow D_T^\top A_T$ ▷ $C_T \in \mathbb{R}^{\|D_T\|_0 \times d}$
  **return** $(C_T, D_T)$
**end procedure**

**Lemma C.2.** *Let $B$ be an $\epsilon$-coreset of $A$ and let $p \notin A$, then $B \cup \{p\}$ is an $\epsilon$-coreset of $A \cup \{p\}$.*

*Proof.* Since $B$ is an $\epsilon$-coreset of $A$, we know that for any $x \in X$, $\mathrm{cost}(B, x) = (1 \pm \epsilon) \cdot \mathrm{cost}(A, x)$. We note that

$$\begin{aligned}\mathrm{cost}(B \cup \{p\}, x) &= \mathrm{cost}(B, x) + \mathrm{cost}(\{p\}, x) \\ &\leq (1+\epsilon) \cdot \mathrm{cost}(A, x) + \mathrm{cost}(\{p\}, x) \\ &\leq (1+\epsilon) \cdot \mathrm{cost}(A \cup \{p\}, x),\end{aligned}$$

the lower bound can be established similarly. □

Theorem C.3 is stated for Algorithm 2, the iterative form of the recursive scheme of Algorithm 1 in Section 2.1.

**Theorem C.3.** *Let $A \in \mathbb{R}^{n \times d}$ with $n \geq 2$, let $\epsilon, \delta \in (0,1)$, and let $w$ satisfy Assumption C.1. Then, Algorithm 2 is a quantum algorithm that, with probability at least $1-\delta$, constructs an $\epsilon$-coreset $C$ of $A$. More precisely, let $\mathcal{F}$ be the event that every intermediate sample $C_t$, $1 \leq t \leq T-1$, of Algorithm 2 is a $0.01$-coreset of $A_t$ and that each of its $T+1$ calls of* QSAMPLE *returns an exact sample, with $T$ as in that algorithm. Then $\Pr[\mathcal{F}] \geq 1 - \delta/2$, with probability at least $1-\delta$ the event $\mathcal{F}$ occurs and $C$ is an $\epsilon$-coreset of $A$, and conditioned on $\mathcal{F}$ the expected size of $C$ is $O(s)$, where $s := 4c \cdot g(\epsilon, \delta/2, n, d) \cdot \mathrm{sum}(w)$. Moreover, if a classical oracle for $w_i(Z, Y)$ can be implemented with*

- *Preprocessing in time $\mathcal{T}_{\mathrm{prep}}(|Y|, d)$, linear in $|Y|$ for fixed $n$ and $d$,*
- *Query time $\mathcal{T}_{\mathrm{query}}(|Y|, d)$ for computing $w_i(Z, Y)$ and $w_i(Z, Y \cup \{z_i\})$ for any $i \in [|Z|]$, where $z_i$ denotes the $i$-th point of the finite set $Z \subseteq \mathbb{R}^d$, and we assume that this query time is bounded by $\mathcal{T}_{\mathrm{query}}(s', d)$ for every set $Y$ arising in the algorithm,*

*the algorithm runs, conditioned on $\mathcal{F}$, in expected time*

$$O(1 + \log(n/s')) \cdot \mathcal{T}_{\mathrm{prep}}(s', d) + \widetilde{O}(\sqrt{n(s+s')} \cdot \mathcal{T}_{\mathrm{query}}(s', d)),$$

*where $s' := 4c \cdot g(0.01, \delta', n, d) \cdot \mathrm{sum}(w)$ with $\delta' = \delta/(2\lceil \log_2 n \rceil)$, and the algorithm uses $\widetilde{O}(\sqrt{n(s+s')})$ queries to the rows or columns of $A$, likewise in expectation conditioned on $\mathcal{F}$. In all applications in this paper $s' = \widetilde{O}(s)$, because either $s = s'$ or $g$ scales as $\epsilon^{-2}$ up to logarithmic factors and $\epsilon < 1$. The bounds then read $O(1 + \log(n/s')) \cdot \mathcal{T}_{\mathrm{prep}}(s', d) + \widetilde{O}(\sqrt{ns} \cdot \mathcal{T}_{\mathrm{query}}(s', d))$ and $\widetilde{O}(\sqrt{ns})$ queries.*

*Proof.* We divide the proof into three parts.

**One level.** Let $A$ be a set of $n$ points, let $A' \subset_{1/2} A$, let $\epsilon_0 \in [0, 1/3]$, and let $C'$ be an $\epsilon_0$-coreset of $A'$, where for $\epsilon_0 = 0$ we take $C' = A'$. We claim that for all $i$,

$$w_i'(A, A'; C') \in I_{\epsilon_0} \cdot w_i'(A, A'), \qquad I_{\epsilon_0} := [(1+\epsilon_0)^{-1}, (1-\epsilon_0)^{-1}], \tag{1}$$

and that

$$\mathbb{E}[\|w'(A, A')\|_1] \leq 2\,\mathrm{sum}(w). \tag{2}$$

For (1), we consider two cases. If $a_i \in A'$, then $w_i'(A, A') = w_i(A, A')$, and by the coreset preserves weights condition of Assumption C.1 with $B = C'$ and $C = A$, we have $w_i'(A, A'; C') = w_i(A, C') \in I_{\epsilon_0} \cdot w_i(A, A') = I_{\epsilon_0} \cdot w_i'(A, A')$. If $a_i \notin A'$, then $C' \cup \{a_i\}$ is an $\epsilon_0$-coreset of $A' \cup \{a_i\}$ by Lemma C.2, and thus the same condition gives $w_i'(A, A'; C') = w_i(A, C' \cup \{a_i\}) \in I_{\epsilon_0} \cdot w_i(A, A' \cup \{a_i\}) = I_{\epsilon_0} \cdot w_i'(A, A')$.

For (2), we first condition on $|A'| = m$, and given this event, $A'$ is a uniformly random subset of $A$ of size $m$. Let $S$ be the diagonal indicator matrix of $A'$, so $SA = A'$. We show that

$$\mathbb{E}[\|w'(A, SA)\|_1 \mid |S| = m] \leq \frac{n+1}{m+1} \cdot \mathrm{sum}(w).$$

Let us denote $S^{(i)}$ as the diagonal indicator matrix for $S \cup \{i\}$. Then, note

$$\sum_{i=1}^{n} w_i'(A, SA) = \sum_{i \in S} w_i(A, SA) + \sum_{i \notin S} w_i(A, S^{(i)}A)$$

$$= \|w(SA, SA)\|_1 + \sum_{i \notin S} w_i(A, S^{(i)}A)$$
$$\leq \mathrm{sum}(w) + \sum_{i \notin S} w_i(A, S^{(i)}A),$$

where the second step uses the convention of Assumption C.1, which we also use below, and the last step uses the consistent total weights assumption. To bound the second term, note that it is generated via the following random process: first selecting $S$, then selecting a random $i \notin S$ and returning $w_i(A, S^{(i)}A)$. Since there are $n - m$ points not in $SA$, the expected value of this process is $\frac{1}{n-m}\,\mathbb{E}[\sum_{i \notin S} w_i(A, S^{(i)}A)]$. The key observation is that this process is equivalent to another process: pick a random subset $S' \subset [n]$ of size $m + 1$, then randomly pick a point $a_i \in S'A$ and return $w_i(A, S'A)$. In expectation, this is equal to the average weight over $S'A$. Since $S'A$ contains $m + 1$ points and by the consistent total weights assumption, the average weight is at most $\frac{\mathrm{sum}(w)}{m+1}$. Therefore,

$$\mathbb{E}[\sum_{i \notin S} w_i(A, S^{(i)}A) \mid |S| = m] \leq (n - m) \cdot \frac{\mathrm{sum}(w)}{m + 1},$$

combining these results, we obtain the following expectation bound:

$$\mathbb{E}[\sum_{i=1}^{n} w'_i(A, SA) \mid |S| = m] \leq \mathrm{sum}(w) + (n - m) \cdot \frac{\mathrm{sum}(w)}{m + 1} = \frac{n + 1}{m + 1} \cdot \mathrm{sum}(w).$$

Since $|A'|$ is a sum of $n$ independent unbiased bits, we have $\Pr[|A'| = m] = \binom{n}{m} 2^{-n}$. Using $\frac{n+1}{m+1}\binom{n}{m} = \binom{n+1}{m+1}$, we obtain

$$\mathbb{E}[\|w'(A, A')\|_1] \leq \mathrm{sum}(w) \cdot \mathbb{E}\left[\frac{n + 1}{|A'| + 1}\right] = \mathrm{sum}(w) \cdot 2^{-n} \sum_{m=0}^{n} \binom{n + 1}{m + 1}$$
$$= (2 - 2^{-n}) \cdot \mathrm{sum}(w) \leq 2\,\mathrm{sum}(w),$$

which is (2).

Next, we derive the consequences of these two bounds. By the uniform sampling bound, we have $w'_i(A, A') \geq w_i(A, A)$. Hence, (1) gives $w'_i(A, A'; C') \geq (1 + \epsilon_0)^{-1}\, w_i(A, A) \geq (1 - \epsilon_0)\, w_i(A, A)$ for all $i$, and on the event that $C'$ is an $\epsilon_0$-coreset of $A'$, it also gives

$$\|w'(A, A'; C')\|_1 \leq \frac{1}{1 - \epsilon_0} \cdot \|w'(A, A')\|_1.$$

Now let $\epsilon_1, \delta_1 \in (0, 1)$ and let $g_1 \geq g(\epsilon_1, \delta_1, n, d)$. In the algorithm, $g$ is evaluated at the size of the input set, and this is admissible by the remarks following Assumption C.1. We sample $A$ with $q_i = \min\{1, c \cdot g_1 \cdot w'_i(A, A'; C')\}$, where $c = 1000$, and we write $q_i = \min\{1, g_1 \cdot u_i\}$ with $u_i := c \cdot w'_i(A, A'; C') \geq c(1 - \epsilon_0)\, w_i(A, A) \geq w_i(A, A)$. By the importance sampling bound of Assumption C.1, the sample is an $\epsilon_1$-coreset of $A$ with probability at least $1 - \delta_1$, and its expected size is at most $g_1 \|u\|_1 = c\, g_1 \|w'(A, A'; C')\|_1$. Taking the expectation over $A'$ and over the randomness that produced $C'$, on the event that $C'$ is an $\epsilon_0$-coreset of $A'$, the expected size is at most

$$c\, g_1 \cdot \frac{1}{1 - \epsilon_0} \cdot 2\,\mathrm{sum}(w) \leq 4c\, g_1 \cdot \mathrm{sum}(w)$$

for $\epsilon_0 \leq 1/3$. For $\epsilon_0 = 0.01$, the factor is $2/0.99 < 2.03$.

**The chain of levels.** For $t \leq T - 1$, let $\mathcal{E}_t$ be the event that $C_t$ is a 0.01-coreset of $A_t$, and let $\mathcal{E}_T$ be the event that $C_T$ is an $\epsilon$-coreset of $A_T = A$. The set $C_0 = A_0$ is an exact coreset of $A_0$, so $\mathcal{E}_0$ holds. By the choice of $T$, we have $\mathbb{E}\,|A_0| = n2^{-T} \leq s'$. Let $\mathcal{G}$ be the event that the $T + 1$ calls of QSAMPLE in the algorithm, for $A_0$ and for $D_1, \ldots, D_T$, all return exact samples. Each call is run with failure probability $\delta'/(T + 1)$, which gives $\Pr[\mathcal{G}^c] \leq \delta'$. Moreover, on $\mathcal{G}$ every sampling step is a Bernoulli sample with the stated probabilities, so the one-level statement applies as written, whereas a failed call may return an arbitrary subset. For $1 \leq t \leq T - 1$, we apply the one-level statement to $A_{t-1} \subset_{1/2} A_t$ with $C' = C_{t-1}$, $\epsilon_0 = 0$ if $t = 1$ and $\epsilon_0 = 0.01$ otherwise,

and $(\epsilon_1, \delta_1) = (0.01, \delta')$. Conditioned on $A_{t-1}$, $C_{t-1}$, and $\mathcal{E}_{t-1}$, the level fails with probability at most $\delta'$, and $\mathbb{E}[|C_t|\,\mathbf{1}_{\mathcal{E}_{t-1}\cap\mathcal{G}}] \le 4c \cdot g(0.01, \delta', n, d) \cdot \mathrm{sum}(w) = s'$. For $t = T$, we apply it with $(\epsilon_1, \delta_1) = (\epsilon, \delta/2)$, so, conditioned on $\mathcal{E}_{T-1}$, the final level fails with probability at most $\delta/2$, and $\mathbb{E}[|C_T|\,\mathbf{1}_{\mathcal{E}_{T-1}\cap\mathcal{G}}] \le 4c \cdot g(\epsilon, \delta/2, n, d) \cdot \mathrm{sum}(w) = s$. Since $s' \ge 1$, we have $T \le \lceil \log_2 n \rceil$ and hence $T\delta' \le \delta/2$. Let $\mathcal{F} := \mathcal{G} \cap \bigcap_{t=1}^{T-1} \mathcal{E}_t$ be the event named in the statement. A union bound over the levels and the sampler calls gives $\Pr[\mathcal{F} \cap \mathcal{E}_T] \ge 1 - (T-1)\delta' - \delta' - \delta/2 \ge 1 - \delta$ and $\Pr[\mathcal{F}] \ge 1 - T\delta' \ge 1 - \delta/2 \ge 1/2$. Since $\mathcal{F} \subseteq \mathcal{E}_{t-1} \cap \mathcal{G}$ for every $t \in [T]$, the bounds above give $\mathbb{E}[|C_t| \mid \mathcal{F}] \le \mathbb{E}[|C_t|\,\mathbf{1}_{\mathcal{E}_{t-1}\cap\mathcal{G}}]/\Pr[\mathcal{F}] \le 2s'$ for $t \le T-1$ and $\mathbb{E}[|C_T| \mid \mathcal{F}] \le 2s$. All expectations in the running time analysis below are conditioned on $\mathcal{F}$. This proves the coreset guarantee and the size bound. Finally, the factor $O(\log((T+1)/\delta')) = O(\log(\log n/\delta))$ in the running time of the sampler calls is absorbed by $\widetilde{O}(\cdot)$.

**Running time.**

- Since we realize the chain by $T$ random bits per point as described at the beginning of this section, testing whether $a_i \in A_t$ costs $O(T) = O(\log n)$ per query, which the $\widetilde{O}$ absorbs. We learn the set $A_0$ explicitly by applying Lemma B.15 to $q_i = \mathbf{1}[a_i \in A_0]$, which takes $\widetilde{O}(\sqrt{n \cdot \mathbb{E}\,|A_0|}) = \widetilde{O}(\sqrt{ns'})$ expected time and queries. Reading its points then takes $|A_0|$ row queries and $O(s'd)$ expected time, which we charge to the first preprocessing. Every preprocessing reads its input, so $\mathcal{T}_{\mathrm{prep}}(s', d) \ge s'd$.

- For each level $t \in [T]$, we first prepare a classical oracle for $w'_i(A_t, A_{t-1}; C_{t-1})$ in $\mathcal{T}_{\mathrm{prep}}(|C_{t-1}|, d)$ time. Moreover, $\mathbb{E}[|C_{t-1}| \mid \mathcal{F}] \le 2s'$ for every $t \in [T]$, where for $t = 1$ this holds because $\mathbb{E}\,|A_0| \le s'$ and $\Pr[\mathcal{F}] \ge 1/2$. Since $\mathcal{T}_{\mathrm{prep}}$ is linear in its first argument by hypothesis, we have $\mathbb{E}[\mathcal{T}_{\mathrm{prep}}(|C_{t-1}|, d) \mid \mathcal{F}] \le \mathcal{T}_{\mathrm{prep}}(2s', d) = O(\mathcal{T}_{\mathrm{prep}}(s', d))$, and we write $\mathcal{T}_{\mathrm{prep}}(s', d)$ for this quantity. The oracles of Lemmas C.8 and C.16 have linear preprocessing time. For the oracle of Lemma C.24, whose preprocessing time is superlinear, the same accounting applies since the proof of Theorem C.25 bounds every level size by $O(s')$ on the success event through Lemma C.22. A query for a point $a_i$ evaluates $w_i(A_t, C_{t-1})$ or $w_i(A_t, C_{t-1} \cup \{a_i\})$ according to the membership bit of $a_i$, which takes $\mathcal{T}_{\mathrm{query}}(s', d)$ time.

- Next, for $t \le T-1$ we sample according to $q_i = \min\{1, c \cdot g(\epsilon', \delta', n, d) \cdot w'_i(A_t, A_{t-1}; C_{t-1})\}$ for $a_i \in A_t$ and $q_i = 0$ for $a_i \notin A_t$, with

  $$\mathbb{E}[\sum_{i \in A_t} q_i \mid \mathcal{F}] \le c \cdot g(\epsilon', \delta', n, d) \cdot \mathbb{E}[\sum_{i \in A_t} w'_i(A_t, A_{t-1}; C_{t-1})\,\mathbf{1}_{\mathcal{E}_{t-1}}]/\Pr[\mathcal{F}] \le\ 1.02\, s'$$

  by the one-level statement, where we use the bound $2\,\mathrm{sum}(w)/0.99$ on the expectation there, the definition $s' = 4c\, g(\epsilon', \delta', n, d)\,\mathrm{sum}(w)$, and $\Pr[\mathcal{F}] \ge 1/2$. Using Lemma B.15 and Jensen's inequality $\mathbb{E}\sqrt{n \sum_i q_i} \le \sqrt{n\ \mathbb{E} \sum_i q_i}$, this step can be implemented in expected time

  $$\widetilde{O}(\sqrt{ns'}) \cdot \mathcal{T}_{\mathrm{query}}(s', d).$$

- Forming $C_t$ requires selecting and weighting $\|D_t\|_0$ points, which can be done in $O(s'd)$ expected time.

- Finally, we do a resampling with $\epsilon$ to form the final coreset. By the same computation as for the intermediate levels, we have $\mathbb{E}[\sum_i q_i \mid \mathcal{F}] \le 1.02\, s$, and this step takes

  $$\mathcal{T}_{\mathrm{prep}}(s', d) + \widetilde{O}(\sqrt{ns} \cdot \mathcal{T}_{\mathrm{query}}(s', d))$$

  expected time. Summing the $T = O(1 + \log(n/s'))$ oracle preparations gives the $O(1 + \log(n/s')) \cdot \mathcal{T}_{\mathrm{prep}}(s', d)$ term. The $T-1$ intermediate sampling rounds together with the final one cost $\widetilde{O}((\sqrt{ns'} + \sqrt{ns}) \cdot \mathcal{T}_{\mathrm{query}}(s', d)) = \widetilde{O}(\sqrt{n(s+s')} \cdot \mathcal{T}_{\mathrm{query}}(s', d))$, where the $\widetilde{O}$ absorbs the factor $T = O(\log n)$. Each oracle query makes one row query, and reading $A_0$ and the sampled points costs $O(T \cdot \min\{s', n\} + \min\{s, n\}) \le \widetilde{O}(\sqrt{n(s+s')})$ further queries in expectation conditioned on $\mathcal{F}$, as $\min\{a, n\} \le \sqrt{an}$. Therefore, the query count is $\widetilde{O}(\sqrt{n(s+s')})$. □

Our main contribution is to prove that *sensitivity sampling* satisfies Assumption C.1.

**Definition C.4.** *Let $A = \{a_1, \dots, a_n\} \subset \mathbb{R}^d$ and let $\mathrm{cost} : \mathbb{R}^d \times X \to \mathbb{R}_{\geq 0}$ be a cost function. We define the sensitivity of $a_i$ with respect to $B$, denoted by $s_i(A, B)$, as*

$$s_i(A, B) = \max_{x \in X, \mathrm{cost}(B,x) \neq 0} \frac{\mathrm{cost}(a_i, x)}{\mathrm{cost}(B, x)}$$

We also need to define the dimension of a system $(A, X, \mathrm{cost})$:

**Definition C.5.** *Given a point set $A = \{a_1, \dots, a_n\} \subset \mathbb{R}^d$, a space $X$ and a cost function $\mathrm{cost} : \mathbb{R}^d \times X \to \mathbb{R}_{\geq 0}$, for a weight vector $v \in \mathbb{R}^n_{\geq 0}$, a subset $S \subseteq [n]$, $x \in X$ and $r \in [0, \infty)$ we define*

$$\mathrm{range}_{S,v}(x, r) = \{i \in S : v_i \cdot \mathrm{cost}(a_i, x) \leq r\}.$$

*The* dimension *of $(A, X, \mathrm{cost})$ is the smallest integer $\dim$ such that for every subset $S \subseteq [n]$ with $|S| \geq 2$ and every weight vector $v \in \mathbb{R}^n_{\geq 0}$ we have*

$$|\{\mathrm{range}_{S,v}(x, r) : x \in X, r \in [0, \infty)\}| \leq |S|^{\dim}.$$

**Lemma C.6** (Sensitivity sampling with independent inclusions)**.** *Let $A = \{a_1, \dots, a_n\} \subset \mathbb{R}^d$, let $w \in \mathbb{R}^n_{\geq 0}$, and write $\mathrm{cost}_w(A, x) = \sum_{i=1}^n w_i \cdot \mathrm{cost}(a_i, x)$. Let $\dim \geq 1$ be an integer that is at least the dimension of $(A, X, \mathrm{cost})$ of Definition C.5. Let $u \in \mathbb{R}^n_{\geq 0}$ satisfy*

$$u_i \geq \sup_{x \in X:\ \mathrm{cost}_w(A,x) \neq 0} \frac{w_i \cdot \mathrm{cost}(a_i, x)}{\mathrm{cost}_w(A, x)} \qquad \text{for all } i \in [n]$$

*(for unit weights, $u_i \geq s_i(A, A)$), let $t \geq \max\{\sum_{i=1}^n u_i, \dim, 2\}$, and let $\epsilon, \delta \in (0, 1)$. Let $c_0$ be a sufficiently large absolute constant, let*

$$g := c_0 \epsilon^{-2} \cdot (\dim \cdot \log(t/\epsilon) + \log(1/\delta)), \qquad q_i := \min\{1, g \cdot u_i\},$$

*and let $S \subseteq [n]$ contain each $i$ independently with probability $q_i$, the point $a_i$ with $i \in S$ receiving the weight $w_i/q_i$. Then with probability at least $1 - \delta$, for all $x \in X$,*

$$\left| \mathrm{cost}_w(A, x) - \sum_{i \in S} \frac{w_i}{q_i} \cdot \mathrm{cost}(a_i, x) \right| \leq \epsilon \cdot \mathrm{cost}_w(A, x).$$

*Moreover, $\mathbb{E}[|S|] = \sum_{i=1}^n q_i \leq g \sum_{i=1}^n u_i$, and $|S| \leq 2gt$ with probability at least $1 - \delta$.*

*Proof.* The counterpart of this statement for a prescribed number of independent draws from the distribution $u/\|u\|_1$ is Theorem 17 of Braverman et al. [2021a] (Theorem 4.1 of Feldman and Langberg [2011]). The sampling primitive of this paper in Lemma B.15 includes each point independently with its own probability, so we prove the statement for that model. Throughout, $\log$ denotes the natural logarithm.

*Size.* Since $\mathbb{E}\,|S| = \sum_i q_i \leq g \sum_i u_i \leq gt$, a Chernoff bound gives $\Pr[|S| \geq 2gt] \leq e^{-gt/3} \leq \delta$, where the last step uses $gt \geq g \geq 3 \log(1/\delta)$.

*Reduction to ranges.* If $\mathrm{cost}_w(A, x) = 0$, both sides vanish. Otherwise, points with $q_i = 1$ contribute equally to both sides, and points with $u_i = 0$ contribute to neither. On the remaining set $L$, we define $h_x(i) := w_i \,\mathrm{cost}(a_i, x)/(q_i \,\mathrm{cost}_w(A, x)) \leq 1/g$ and $\mu(R) := \sum_{i \in R} q_i$. Let $R_x(r) := \{i \in L : h_x(i) > r\}$, which is empty for $r \geq 1/g$, and let $f_x(i) := w_i \,\mathrm{cost}(a_i, x)/\mathrm{cost}_w(A, x) = q_i h_x(i)$, so $\sum_{i \in L} f_x(i) \leq 1$. By the layer-cake formula, we have $\sum_{i \in S \cap L} h_x(i) = \int_0^{1/g} |S \cap R_x(r)|\,\mathrm{d}r$ and $\sum_{i \in L} q_i h_x(i) = \int_0^{1/g} \mu(R_x(r))\,\mathrm{d}r$. Therefore, it suffices to show that $||S \cap R| - \mu(R)| \leq \frac{\epsilon}{2} \max\{\mu(R), g\}$ for every level set $R = R_x(r)$, since integrating this bound shows that the error is at most $\frac{\epsilon}{2}(\sum_{i \in L} f_x(i) + g \cdot \frac{1}{g}) \leq \epsilon$. By the hypothesis on $\dim$, these sets have at most $|M|^{\dim}$ traces on every $M \subseteq L$ with $|M| \geq 2$.

*Symmetrization.* Let $S'$ be an independent copy of $S$ and let $m_R := \max\{\mu(R), g\}$. Since $\mathrm{Var}|S' \cap R| \leq \mu(R)$, Chebyshev's inequality gives $\Pr[||S' \cap R| - \mu(R)| \leq \frac{\epsilon}{4} m_R] \geq 1 - 16/(\epsilon^2 g) \geq 1/2$ once $\epsilon^2 g \geq 32$. The first violating $R$ in a fixed enumeration is a function of $S$ alone, so it suffices

to bound $|S \cap R| - |S' \cap R|$ on the event that $S'$ is well behaved, at the price of a factor 2. Conditionally on $S \cup S'$ and $S \triangle S'$, the differences of the indicators on $S \triangle S'$ are independent uniform signs. Let $U := S \cup S'$, $M := S \triangle S'$, and $c_R := |S \cap R| + |S' \cap R| \geq |M \cap R|$. Both $c_R$ and $M \cap R$ depend on $R$ only through the trace $R \cap U$. First, by Markov's inequality, $\Pr[|S' \cap R| \leq 2m_R] \geq 1/2$ for each $R$. Therefore, as above, if some level set has $|S \cap R| > 4m_R$, then with probability at least $1/2$ some $R$ has $|S \cap R| > 4m_R \geq 4g$ and $|S' \cap R| \leq 2m_R$. Such an $R$ has $c_R > 4g$ and $\sum_{i \in M \cap R} \sigma_i = |S \cap R| - |S' \cap R| > |S \cap R|/2 \geq c_R/3$, so Hoeffding's inequality gives conditional probability at most $e^{-c_R/18} \leq e^{-2g/9} \leq e^{-\epsilon^2 g/192}$ per trace. Hence, with $N \leq (4gt)^{\dim}$ traces and the condition displayed below, every level set has $|S \cap R| \leq 4m_R$ except with probability $2(\delta/16 + \delta/32) = 3\delta/16$, where the term $\delta/16$ accounts for $\Pr[|S| > 2gt \text{ or } |S'| > 2gt] \leq 2e^{-gt/3}$. Second, suppose this event holds. Then a level set $R$ with $||S \cap R| - \mu(R)| > \frac{\epsilon}{2} m_R$ and $||S' \cap R| - \mu(R)| \leq \frac{\epsilon}{4} m_R$ has $|S' \cap R| \leq 2m_R$, and hence $|M \cap R| \leq c_R \leq 6m_R$ and $|\sum_{i \in M \cap R} \sigma_i| > \frac{\epsilon}{4} m_R$. Since $\mu(R)$ is not determined by the trace $T = R \cap U$, we instead define $m_T^\star := \max\{g, |M \cap T|/6\}$. Such an $R$ has $m_R \geq m_T^\star$, so $|\sum_{i \in M \cap T} \sigma_i| > \frac{\epsilon}{4} m_T^\star$ with $|M \cap T| \leq 6m_T^\star$, and Hoeffding's inequality gives conditional probability at most $2e^{-\epsilon^2 m_T^\star/192} \leq 2e^{-\epsilon^2 g/192}$ per trace. If $|U| \leq 1$, then the sign sums are at most 1 in absolute value while both thresholds exceed 1, so neither event occurs. There are at most $(4gt)^{\dim}$ traces when $|S|, |S'| \leq 2gt$, and a union bound over them gives $Ne^{-\epsilon^2 g/192} \leq \delta/32$. Hence, with the outer symmetrization factor 2, the total failure probability is at most $2(3\delta/16 + \delta/16 + 2 \cdot \delta/32) = 5\delta/8 \leq \delta$ once

$$\frac{\epsilon^2 g}{192} \geq \dim \cdot \log(4gt) + \log(32/\delta).$$

Let $D := \dim \cdot \log(t/\epsilon) + \log(1/\delta)$, then $g = c_0 \epsilon^{-2} D$. To check the condition, we expand $\log(4gt)$ and use $\log(a+b) \leq \log 2 + \log^+ a + \log^+ b$ with $\log^+ y := \max\{0, \log y\}$, $\dim \leq t$, and $\ell \log^+ b \leq \ell \log \ell + b$ for $\ell \geq 1$. A routine computation then shows that the right-hand side is at most $(\log(4c_0)/\log 2 + 12) \cdot D$, so the condition holds once $c_0 \geq 192(\log(4c_0)/\log 2 + 12)$. Moreover, the same choice gives $\epsilon^2 g \geq 32$ for the Chebyshev step, $g \geq 3\log(1/\delta)$ for the size bound, and $gt \geq 3\log(32/\delta)$ for the event $|S|, |S'| \leq 2gt$. □

**Theorem C.7.** *Let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$ and let $\mathrm{cost} : \mathbb{R}^d \times X \to \mathbb{R}_{\geq 0}$. Moreover, suppose the total sensitivity has a finite upper bound, i.e., there exists some $\mathrm{sum}(s) < \infty$ such that for any finite subset $C \subset \mathbb{R}^d$, $\sum_{i \in C} s_i(C, C) \leq \mathrm{sum}(s)$. Then, the sensitivity of $A$ with respect to $B$, $s(A, B)$ (Def. C.4) satisfies Assumption C.1.*

*Proof.* We need to prove $s(A, B)$ satisfies the four items in Assumption C.1.

- Consistent total weights: by assumption, we have that for any $S \subseteq [n]$, $\sum_{i \in S} s_i(A_S, A_S) \leq \mathrm{sum}(s)$ with $\mathrm{sum}(s)$ being finite.
- Uniform sampling bound: we analyze by cases. For the first case, where $a_i \in A'$, we have $w_i'(A, A') = s_i(A, A')$. Let $x_1, x_2$ be the two points that realize $s_i(A, A')$ and $s_i(A, A)$, respectively. Suppose $\mathrm{cost}(A', x_2) \neq 0$, then

$$\begin{aligned} \frac{\mathrm{cost}(a_i, x_1)}{\mathrm{cost}(A', x_1)} &\geq \frac{\mathrm{cost}(a_i, x_2)}{\mathrm{cost}(A', x_2)} \\ &\geq \frac{\mathrm{cost}(a_i, x_2)}{\mathrm{cost}(A', x_2) + \mathrm{cost}(A \setminus A', x_2)} \\ &= \frac{\mathrm{cost}(a_i, x_2)}{\mathrm{cost}(A, x_2)} \end{aligned}$$

  where we use the fact that cost is nonnegative, therefore increasing the denominator will only decrease the fraction. On the other hand, if $\mathrm{cost}(A', x_2) = 0$, then it must be that $\mathrm{cost}(a_i, x_2) = 0$ due to the nonnegativity of cost. Hence, $s_i(A, A) = 0$, and consequently $s_i(A, A') = 0$ as otherwise we could pick $x_1$ for $s_i(A, A)$.

  For the next case, where $a_i \notin A'$, we have $w_i'(A, A') = s_i(A, A' \cup \{a_i\})$. Again, let $x_1, x_2$ be the two points that realize $s_i(A, A' \cup \{a_i\})$ and $s_i(A, A)$. The argument is similar:

suppose $\mathrm{cost}(A', x_2) \neq 0$, then

$$\begin{aligned}\frac{\mathrm{cost}(a_i, x_1)}{\mathrm{cost}(A', x_1) + \mathrm{cost}(a_i, x_1)} &\geq \frac{\mathrm{cost}(a_i, x_2)}{\mathrm{cost}(A', x_2) + \mathrm{cost}(a_i, x_2)} \\ &\geq \frac{\mathrm{cost}(a_i, x_2)}{\mathrm{cost}(A', x_2) + \mathrm{cost}(a_i, x_2) + \mathrm{cost}(A \setminus (A' \cup \{a_i\}), x_2)} \\ &= \frac{\mathrm{cost}(a_i, x_2)}{\mathrm{cost}(A, x_2)}.\end{aligned}$$

If $\mathrm{cost}(A', x_2) = 0$ and $\mathrm{cost}(a_i, x_2) \neq 0$, then $\mathrm{cost}(A' \cup \{a_i\}, x_2) = \mathrm{cost}(a_i, x_2) \neq 0$, so $x_2$ is admissible in the definition of $s_i(A, A' \cup \{a_i\})$ and attains the ratio $\mathrm{cost}(a_i, x_2)/(\mathrm{cost}(A', x_2) + \mathrm{cost}(a_i, x_2)) = 1$. Since every sensitivity of a point of the reference set is at most 1, this gives $s_i(A, A' \cup \{a_i\}) = 1 \geq s_i(A, A)$. If instead $\mathrm{cost}(a_i, x_2) = 0$, then $s_i(A, A) = \mathrm{cost}(a_i, x_2)/\mathrm{cost}(A, x_2) = 0 \leq s_i(A, A' \cup \{a_i\})$.

- Importance sampling bound: let $\dim$ be the dimension of $(A, X, \mathrm{cost})$ of Definition C.5, which is the quantity required by Lemma C.6. Note that $\dim \leq n$, since a set of $m \geq 2$ points has at most $2^m \leq m^n$ subsets and we assume $n \geq 2$. Let $u \in \mathbb{R}^n$ satisfy $u_i \geq s_i(A, A)$ for all $i$, and let $\overline{u}_i := \min\{1, u_i\}$, which still satisfies $\overline{u}_i \geq s_i(A, A)$ since $s_i(A, A) \leq 1$ for every $i$. Moreover, $\sum_i \overline{u}_i \leq n$, and $\min\{1, g \cdot u_i\} = \min\{1, g \cdot \overline{u}_i\}$ for every $g \geq 1$. We apply Lemma C.6 to $\overline{u}$ with unit weights and $t := n \geq \max\{\sum_i \overline{u}_i, \dim, 2\}$. Hence, with

$$g(\epsilon, \delta, n, d) = c_0 \epsilon^{-2} \cdot (\dim \cdot \log(n/\epsilon) + \log(1/\delta)) \geq 1$$

and $q_i = \min\{1, g(\epsilon, \delta, n, d) \cdot u_i\} = \min\{1, g(\epsilon, \delta, n, d) \cdot \overline{u}_i\}$, the weighted sample is an $\epsilon$-coreset of $A$ with probability at least $1 - \delta$, and its expected size is at most $g(\epsilon, \delta, n, d) \cdot \|\overline{u}\|_1 \leq g(\epsilon, \delta, n, d) \cdot \|u\|_1$. This verifies the importance sampling bound for every overestimate $u$, as required by Assumption C.1.

- Coreset preserves weights: let $B$ be an $\epsilon$-coreset of $A$. Then, we know that for any $x \in X$, $\mathrm{cost}(B, x) = (1 \pm \epsilon) \cdot \mathrm{cost}(A, x)$. Now, let $C \subset \mathbb{R}^d$ be any fixed set of points, and let $x_1, x_2 \in X$ be the points that achieve $s_i(C, A)$ and $s_i(C, B)$. We have:

$$\begin{aligned}w_i(C, B) &= s_i(C, B) \\ &= \frac{\mathrm{cost}(c_i, x_2)}{\mathrm{cost}(B, x_2)} \\ &\leq (1 - \epsilon)^{-1} \cdot \frac{\mathrm{cost}(c_i, x_2)}{\mathrm{cost}(A, x_2)} \\ &\leq (1 - \epsilon)^{-1} \cdot \frac{\mathrm{cost}(c_i, x_1)}{\mathrm{cost}(A, x_1)} \\ &= (1 - \epsilon)^{-1} \cdot s_i(C, A),\end{aligned}$$

where we use $\mathrm{cost}(B, x_2) \geq (1 - \epsilon) \cdot \mathrm{cost}(A, x_2)$ in the third step. We could similarly establish that $s_i(C, B) \geq (1 + \epsilon)^{-1} \cdot s_i(C, A)$. This proves the assertion. □

In what follows, we demonstrate how to concretely implement sensitivity sampling for various cost functions, such as $\ell_p$ sensitivity and $k$-subspace sensitivity.

### C.1 $\ell_2$ Sensitivity and Leverage Score

Let $X = \mathbb{R}^d$ and $\mathrm{cost}(a_i, x) = (a_i^\top x)^2$. In this case, the $\ell_2$ sensitivity defined as

$$s_i(A, B) = \max_{x \in \mathbb{R}^d, Bx \neq 0} \frac{(a_i^\top x)^2}{\|Bx\|_2^2}$$

is, in fact, the leverage score $\tau_i(A)$. The leverage score has many favorable structures: for example, to obtain an $\epsilon$-coreset, it is sufficient to sample $O(\epsilon^{-2} d \log d)$ points, and one could sample with $w_i(A, A')$ instead of $w_i'(A, A')$.

**Algorithm 3** Classical oracle for leverage score.

**data structure** LEVERAGESCORE
**members**
  $A \in \mathbb{R}^{n \times d}$
  $C \in \mathbb{R}^{s \times d}$
  $M \in \mathbb{R}^{O(\log n) \times d}$
  $N \in \mathbb{R}^{O(\log n) \times d}$
**end members**

**procedure** PREPROCESS($A \in \mathbb{R}^{n \times d}, C \in \mathbb{R}^{s \times d}$)
  $c \leftarrow 1000$
  Compute the compact SVD of $C$: $C = U \Sigma V^\top$ $\triangleright$ $r = \mathrm{rank}(C) \leq s$, $\Sigma \in \mathbb{R}^{r \times r}$ invertible, $V \in \mathbb{R}^{d \times r}$
  Let $G \in \mathbb{R}^{c \log n \times r}$ and $G' \in \mathbb{R}^{c \log n \times d}$ be random Gaussian matrices
  $M \leftarrow G \Sigma^{-1} V^\top$ $\triangleright$ $M \in \mathbb{R}^{c \log n \times d}$
  $N \leftarrow G'(I - VV^\top)$ $\triangleright$ $N \in \mathbb{R}^{c \log n \times d}$, sketch of the orthogonal complement of $\mathrm{span}(C^\top C)$
**end procedure**

**procedure** QUERY($i \in [n]$)
  **if** $Na_i \neq 0$ **then return** $\infty$ $\triangleright$ $a_i \notin \mathrm{span}(C^\top C)$, so $q_i = 1$
  **return** $\|Ma_i\|_2^2$
**end procedure**
**end data structure**

**Lemma C.8.** *Let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$ and define $\mathrm{cost} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ by $\mathrm{cost}(a_i, x) = (a_i^\top x)^2$, and let $w(A, B)$ be defined as*

$$w_i(A, B) = \begin{cases} a_i^\top (B^\top B)^\dagger a_i, & \text{if } a_i \in \mathrm{span}(B^\top B), \\ \infty, & \text{otherwise.} \end{cases}$$

*Then, the weights $w$ satisfy Assumption C.1. Moreover, there exists a randomized algorithm (Algorithm 3) that implements* PREPROCESS *and* QUERY *procedures, with*

- $\mathcal{T}_{\mathrm{prep}}(s, d) = \widetilde{O}(sd^{\omega-1})$.
- $\mathcal{T}_{\mathrm{query}}(s, d) = \widetilde{O}(d)$.

*Proof.* While leverage score is $\ell_2$ sensitivity and we could directly use Theorem C.7, we include a proof that utilizes the structure of leverage score for completeness.

- Consistent total weights: first note that

$$\begin{aligned} \sum_{i=1}^n w_i(A, A) &= \sum_{i=1}^n a_i^\top (A^\top A)^\dagger a_i \\ &= \mathrm{tr}[(A^\top A)^\dagger A^\top A] \\ &= \mathrm{rank}(A) \\ &\leq d \end{aligned}$$

  hence we have $\mathrm{sum}(w) = d$. Let $S \subset [n]$ with $|S| \geq d$, then

$$\begin{aligned} \sum_{i \in S} w_i(A_S, A_S) &= \sum_{i \in S} a_i^\top (A_S^\top A_S)^\dagger a_i \\ &= \mathrm{tr}[(A_S^\top A_S)^\dagger (A_S^\top A_S)] \\ &= \mathrm{rank}(A_S) \\ &\leq d. \end{aligned}$$

- Uniform sampling bound: the proof closely follows that of Theorem 1 of Cohen et al. [2015b], and we analyze by cases. Let $S$ be an indicator matrix with $A' = SA$ and let $S^{(i)}$ be the indicator matrix for $S \cup \{i\}$. We will show that $w'_i(A, A') = w_i(A, S^{(i)}A)$ via case analysis. If $a_i \in A'$, then $w'_i(A, A') = w_i(A, A')$ and $S = S^{(i)}$, consequently $w_i(A, SA) = w_i(A, S^{(i)}A)$. If $a_i \notin A'$, then $w'_i(A, A') = w_i(A, A' \cup \{i\}) = w_i(A, S^{(i)}A)$. This completes the proof. To show the overestimate, observe that $S^{(i)}$ is an indicator matrix for the sample and thus $S^{(i)} \preceq I_n$, we can then conclude

$$A^\top (S^{(i)})^2 A \preceq A^\top A$$

and

$$\begin{aligned} w'_i(A, A') &= a_i^\top (A^\top (S^{(i)})^2 A)^\dagger a_i \\ &\geq a_i^\top (A^\top A)^\dagger a_i \\ &= w_i(A, A). \end{aligned}$$

- Importance sampling bound: this is standard as $w_i(A, A)$ is the leverage score of matrix $A$. Let $g(\epsilon, \delta, n, d) := c\epsilon^{-2} \log(d/\delta)$ for a sufficiently large constant $c$. We include each row independently with probability $q_i = \min\{1, g \cdot u_i\}$ for overestimates $u_i \geq \tau_i(A)$ and reweight it by $1/q_i$. By the matrix Chernoff bound, with probability at least $1 - \delta$ this gives $(1 \pm \epsilon)A^\top A$ in the Loewner order, hence an $\epsilon$-coreset, and the expected size of the sample is $g\|u\|_1$. Lemma 4 of Cohen et al. [2015b] is the same statement with $\log d$ in place of $\log(d/\delta)$ and failure probability $d^{-\Omega(1)}$.

- Coreset preserves weights: because $B$ is an $\epsilon$-coreset of $A$, we know that for any $x \in \mathbb{R}^d$, $\|Bx\|_2^2 = (1 \pm \epsilon) \cdot \|Ax\|_2^2$. This implies that $B^\top B$ is a spectral approximation to $A^\top A$ and $\ker(A) = \ker(B)$. By the monotonicity of the pseudo-inverse on the common range, the same holds for $(B^\top B)^\dagger$ with respect to $(A^\top A)^\dagger$ with the factors $(1 + \epsilon)^{-1}$ and $(1 - \epsilon)^{-1}$. Let $C \subset \mathbb{R}^d$ be any fixed subset of $\mathbb{R}^d$. We conclude the proof by spectral approximation:

$$(1 + \epsilon)^{-1} \cdot c_i^\top (A^\top A)^\dagger c_i \leq c_i^\top (B^\top B)^\dagger c_i \leq (1 - \epsilon)^{-1} \cdot c_i^\top (A^\top A)^\dagger c_i.$$

Now, we turn to the runtime analysis of Algorithm 3. Let $C = U\Sigma V^\top$ be the compact SVD of $C$, with $r = \operatorname{rank}(C)$, $\Sigma \in \mathbb{R}^{r \times r}$ invertible, and $V \in \mathbb{R}^{d \times r}$. Then we have $(C^\top C)^\dagger = (V\Sigma^2 V^\top)^\dagger = V\Sigma^{-2}V^\top$. By definition, for $a_i \in \operatorname{span}(V)$,

$$\begin{aligned} w_i(A, C) &= a_i^\top (C^\top C)^\dagger a_i \\ &= a_i^\top V \Sigma^{-2} V^\top a_i \\ &= \|\Sigma^{-1} V^\top a_i\|_2^2, \end{aligned}$$

and by a standard Johnson-Lindenstrauss trick [Spielman and Srivastava, 2011], it is sufficient to prepare the matrix $G\Sigma^{-1}V^\top$ for a JL matrix $G$. Then, with high probability, all $w_i(A, C)$ are approximated within a factor of $1 \pm \epsilon$, and a simple scaling turns these estimates into overestimates for all $a_i \in \operatorname{span}(C^\top C) = \operatorname{span}(V)$. By definition, the points outside this span have weight $\infty$, and thus $q_i = \min\{1, w_i\} = 1$. This is the convention of Cohen et al. [2015b] under which Lemma C.10 is stated, so its size bound accounts for these points. For such $a_i$ we have $(I - VV^\top)a_i \neq 0$, and for a Gaussian $G'$ the vector $Na_i = G'(I - VV^\top)a_i$ is nonzero with probability 1, while every $a_i \in \operatorname{span}(V)$ satisfies $Na_i = 0$ exactly. Therefore, the test in QUERY returns $\infty$ on the points outside the span and only on them. We assume that this exact zero test on $O(\log n)$ numbers can be performed, and with rational inputs we fingerprint $a_i$ against a rational basis of the orthogonal complement of the row space of $C$, computed by Gaussian elimination at the same $O(sd^{\omega-1})$ cost. It remains to analyze the runtime.

- Computing the compact SVD of $C$ takes $O(sd^{\omega-1})$ time.

- Computing $G\Sigma^{-1}$ takes $\widetilde{O}(s \log n)$ time and then we multiply $G\Sigma^{-1}$ with $V^\top$ which takes $\widetilde{O}(sd)$ time. Computing $N = G' - (G'V)V^\top$ takes $\widetilde{O}(rd) \leq \widetilde{O}(sd)$ time.

- For query, note that $M, N \in \mathbb{R}^{O(\log n)\times d}$, and thus testing $Na_i \neq 0$ and computing $\|Ma_i\|_2^2$ takes $\widetilde{O}(d)$ time.

This completes the proof of the assertion. □

**Remark C.9.** *If we faithfully execute the framework of Theorem C.7, then we would need to compute the $w_i(A, C \cup \{a_i\})$ instead of $w_i(A, C)$. Instead, we only need to sample with $w_i(A, C)$, which is a key feature for leverage score and related notions. We take the same route for Lewis weights in Theorem C.18 and for ridge leverage scores in Theorem C.25. We run the recursion of Algorithm 2 with the one-sided values $w_i(A_t, C_{t-1})$ in place of $w'_i(A_t, A_{t-1}; C_{t-1})$. The coreset guarantee and the level sizes for this sampling rule are proved in Lemmas C.10, C.14, and C.22. We use Theorem C.3 only for its time and query accounting, and thus the oracles of Lemmas C.8, C.16, and C.24 only need to evaluate $w_i(Z, Y)$. Nevertheless, for leverage scores the augmented value is available at no extra cost by the Sherman–Morrison formula of the next remark.*

**Lemma C.10** (Theorem 4 of Cohen et al. [2015b])**.** *Let $A = \{a_1, \dots, a_n\} \subset \mathbb{R}^d$, $m \in [n]$ and $\epsilon, \delta \in (0, 1)$. Form $SA$ by including each point of $A$ independently with probability $\frac{m}{n}$, and let $w$ be as in Lemma C.8. Let $q_i = \min\{1, c\epsilon^{-2}\log(d/\delta) \cdot w_i(A, SA)\}$ for a sufficiently large constant $c$, and let $B$ contain each $a_i$ independently with probability $q_i$, weighted by $1/q_i$. Then $\mathbb{E}[\sum_{i=1}^n q_i] = O(\epsilon^{-2}\frac{nd}{m}\log(d/\delta))$, and with probability at least $1-\delta$, $B$ is an $\epsilon$-coreset of $A$, i.e., $(1-\epsilon)A^\top A \preceq B^\top B \preceq (1+\epsilon)A^\top A$.*

**Remark C.11.** *The statement combines two facts. First, by the monotonicity of the pseudo-inverse as in the proof of Lemma C.8, we have $w_i(A, SA) \geq w_i(A, A) = \tau_i(A)$ for every $i$, and thus $q_i \geq \min\{1, c\epsilon^{-2}\log(d/\delta) \cdot \tau_i(A)\}$. Then, for every realization of $SA$, the spectral approximation follows from the matrix Chernoff bound for leverage score sampling with independent inclusions. This is the bound behind Lemma 4 of Cohen et al. [2015b], and here its oversampling factor $c\epsilon^{-2}\log(d/\delta)$ replaces the $\log d$ and failure probability $d^{-\Omega(1)}$ printed there. Second, we have $\mathbb{E}[\sum_i \min\{1, w_i(A, SA)\}] = O(nd/m)$. Indeed, for $a_i \in \operatorname{span}(SA)$ the Sherman–Morrison formula gives $w_i(A, SA\cup\{a_i\}) = w_i(A, SA)/(1+w_i(A, SA))$, and $w_i(A, SA\cup\{a_i\}) = 1$ otherwise, so in both cases $\min\{1, w_i(A, SA)\} \leq 2\, w'_i(A, SA)$. Moreover, $\mathbb{E}\,\|w'(A, SA)\|_1 \leq 2\frac{n}{m} \cdot d$ by the exchange argument of the proof of Theorem C.3, which applies to independent inclusions as well as to a uniformly random subset of exactly $m$ points.*

Setting $m = n/2$, Lemma C.10 itself is sufficient to prove Theorem C.3, without resorting to $w_i(A, C \cup \{a_i\})$. The factor 2 above is absorbed by the constant $c$. Our following theorem recovers the main result of Apers and Gribling [2026].

**Algorithm 4** Quantum leverage score sampling with a size parameter.

1: **procedure** QLS($A \in \mathbb{R}^{n\times d}, s, \delta$)
2: $s' \leftarrow c \cdot d\log(d\lceil\log_2 n\rceil/\delta)$ ▷ $c$ a sufficiently large constant, and $s' = \widetilde{O}(d)$
3: $(C_T, D_T) \leftarrow$ QITERATESAMPLE($A, 0.01, \delta/4$) with the oracle of Algorithm 3, sampling with $w_i(A_t, C_{t-1})$ in place of $w'_i$ (Lemma C.10), and $O(s')$ points per level in expectation ▷ Algorithm 2
4: LEVERAGESCORE.PREPROCESS($A, C_T$) ▷ Algorithm 3 on $C_T$, $\mathcal{T}_{\mathrm{prep}}(s', d) = \widetilde{O}(d^\omega)$ time
5: $\widetilde{\tau}_i \leftarrow$ LEVERAGESCORE.QUERY$(i)/((1-0.01)(1-\epsilon_{\mathrm{JL}}))$ ▷ evaluated on demand, and $\tau_i(A) \leq \widetilde{\tau}_i \leq O(1) \cdot \tau_i(A)$
6: $q_i \leftarrow \min\{1, s \cdot \widetilde{\tau}_i/d\}$ ▷ $\sum_{i=1}^n q_i \leq s\sum_i \widetilde{\tau}_i/d = O(s)$
7: $D \leftarrow$ QSAMPLE$(q, \delta/(8(T+2)))$ ▷ Lemma B.15
8: **return** $(D^\top A, D)$
9: **end procedure**

In Algorithm 4, $T$ is the number of levels of the recursion in line 3.

**Theorem C.12.** *Let $A = \{a_1, \dots, a_n\} \subset \mathbb{R}^d$ and $\epsilon, \delta \in (0, 1)$. Then, there exists a quantum algorithm that with probability $1-\delta$, constructs an $\epsilon$-coreset $B$ of $A$ of size $O(\epsilon^{-2}d\log(d/\delta))$, in expected time $\widetilde{O}(\epsilon^{-1}n^{0.5}d^{1.5} + d^\omega)$ and $\widetilde{O}(\epsilon^{-1}n^{0.5}d^{0.5})$ queries to points in $A$.*

*Furthermore, for a size parameter $s \geq d$, the procedure* QLS$(A, s, \delta)$ *of Algorithm 4 with probability at least $1-\delta$ computes overestimates $\widetilde{\tau}_i$ of the leverage scores with $\tau_i(A) \leq \widetilde{\tau}_i \leq O(1) \cdot \tau_i(A)$ for*

*every $i \in [n]$, hence $\sum_{i=1}^{n} \widetilde{\tau}_i = O(\mathrm{rank}(A))$, and returns a sample in which point $i$ is included independently with probability $q_i = \min\{1, s \cdot \widetilde{\tau}_i/d\}$ and carries the weight $1/q_i$. The sample has expected size $\sum_i q_i = O(s)$, and size $O(s + \log(1/\delta))$ with probability at least $1 - \delta$. The procedure runs in expected time $\widetilde{O}(n^{0.5}s^{0.5}d + d^{\omega})$ and makes $\widetilde{O}(n^{0.5}s^{0.5})$ queries to points in $A$, the expectations being taken as in Theorem C.3.*

*Proof.* We first prove the statement about QLS and then obtain the $\epsilon$-coreset by a choice of $s$. Let $T = O(\log n)$ be the number of levels of the constant-accuracy recursion and let $s' = O(d \log(dT/\delta)) = \widetilde{O}(d)$ be its expected number of points per level.

*Step 1: the constant-accuracy recursion.* We run QITERATESAMPLE$(A, 0.01, \delta/4)$ with the oracle of Algorithm 3. At level $t$, we sample with $q_i = \min\{1, c\, g \cdot w_i(A_t, C_{t-1})\}$ in place of the augmented weights, where $g = \Theta(\log(dT/\delta))$ is the oversampling factor of Lemma C.10 at accuracy $0.01$ and failure probability $\delta/(8T)$. The per-level failure probability $\delta/(8\lceil \log_2 n \rceil)$ of the call is at most this. For the weights of Lemma C.8, items 1 and 4 of Assumption C.1 are proved there, and we replace items 2 and 3 by Lemma C.10 with $m = n/2$, applied to the Bernoulli$(1/2)$ half $A_{t-1}$ of $A_t$. At level $t$, the oracle evaluates $w_i(A_t, C_{t-1})$. If $C_{t-1}$ is a $0.01$-coreset of $A_{t-1}$, then by item 4 this weight is within a constant factor of $w_i(A_t, A_{t-1})$, and the two are infinite for the same $i$ since $\mathrm{span}(C_{t-1}) = \mathrm{span}(A_{t-1})$. Hence, in this case, the sampling probabilities dominate those of Lemma C.10 and the level produces a $0.01$-coreset $C_t$ of $A_t$ with probability at least $1 - \delta/(8T)$. Moreover, $\sum_{i \in A_t} q_i$ then has expectation $O(s')$ over the uniform half $A_{t-1}$ by the second fact stated after Lemma C.10. Each oracle is prepared with a Gaussian sketch of $O(\log(n/\delta))$ rows, whose distortion bound holds for all rows of $A_t$ with probability at least $1 - \delta/(8T)$. The $T + 1$ calls of QSAMPLE are run with failure probability $\delta/(8(T + 2))$ each, in place of the parameter $\delta'/(T + 1)$ printed in Algorithm 2, and either value closes the union bound below. By a union bound over the $T$ levels, the $T$ sketches, and the $T + 1$ sampler calls, with probability at least $1 - 3\delta/8 - \delta/8 = 1 - \delta/2$, every level succeeds and $C_T$ is a $0.01$-coreset of $A$. Conditioned on this event, every intermediate sample $C_t$, and hence every set on which an oracle is prepared, has expected size $O(s')$ as in Theorem C.3. By the analysis of Theorem C.3 and Lemma C.8, this step takes $O(\log n) \cdot \mathcal{T}_{\mathrm{prep}}(s', d) + \widetilde{O}(\sqrt{ns'} \cdot \mathcal{T}_{\mathrm{query}}(s', d)) = \widetilde{O}(d^{\omega} + n^{0.5}d^{1.5})$ expected time and $\widetilde{O}(\sqrt{ns'}) = \widetilde{O}(n^{0.5}d^{0.5})$ queries.

*Step 2: the additional oracle.* Since $C_T$ is a $0.01$-coreset of $A$, we have $0.99\, A^{\top}A \preceq C_T^{\top}C_T \preceq 1.01\, A^{\top}A$, and thus $\ker(C_T) = \ker(A)$. Hence, by the monotonicity of the pseudo-inverse on the common range, $\tau_i(A)/1.01 \le w_i(A, C_T) \le \tau_i(A)/0.99$ for every $i$. This is item 4 of Assumption C.1 for the weights of Lemma C.8, with $B = C_T$ and $C = A$. When prepared on $C_T$, the procedure QUERY of Algorithm 3 returns $(1 \pm \epsilon_{\mathrm{JL}})\, w_i(A, C_T)$ for all $i$ simultaneously with probability $1 - n^{-\Omega(1)}$. Here $\epsilon_{\mathrm{JL}} \le 1/5$ is the distortion of its Gaussian sketch with $O(\log n)$ rows, and for $\delta < n^{-\Omega(1)}$ we take $O(\log(n/\delta))$ rows. Hence, $\widetilde{\tau}_i := \mathrm{QUERY}(i)/(0.99(1 - \epsilon_{\mathrm{JL}}))$ satisfies

$$\tau_i(A) \le \widetilde{\tau}_i \le \frac{1 + \epsilon_{\mathrm{JL}}}{0.99^2(1 - \epsilon_{\mathrm{JL}})} \cdot \tau_i(A) \le 2\tau_i(A) \qquad \text{for all } i \in [n],$$

and $\sum_i \widetilde{\tau}_i \le 2\,\mathrm{rank}(A) \le 2d$. Consequently, $\sum_i q_i \le s \sum_i \widetilde{\tau}_i/d \le 2s$, and by a Chernoff bound, $|D| \le 2\max\{2s, 3\log(4/\delta)\}$ with probability $1 - \delta/4$. Preparing this oracle takes $\mathcal{T}_{\mathrm{prep}}(s', d) = \widetilde{O}(s'd^{\omega-1}) = \widetilde{O}(d^{\omega})$ time. We then sample using Lemma B.15 with failure probability $\delta/(8(T+2))$, which takes $\widetilde{O}(\sqrt{ns}) \cdot \mathcal{T}_{\mathrm{query}}(s', d) = \widetilde{O}(n^{0.5}s^{0.5}d)$ time and $\widetilde{O}(\sqrt{ns})$ queries. Altogether, QLS succeeds with probability at least $1 - \delta$, since it fails with probability $\delta/2$ in Step 1 and at most $\delta/4 + \delta/8 + \delta/8$ in the size bound, the sampler call, and the sketch of Step 2. It runs in expected time $\widetilde{O}(n^{0.5}d^{1.5} + d^{\omega} + n^{0.5}s^{0.5}d) = \widetilde{O}(n^{0.5}s^{0.5}d + d^{\omega})$ and makes $\widetilde{O}(n^{0.5}(d^{0.5} + s^{0.5}))$ queries, which are the stated bounds for $s \ge d$.

*Step 3: the $\epsilon$-coreset.* We run QLS$(A, s, \delta/2)$ with $s = C\epsilon^{-2}d\log(d/\delta)$ for a sufficiently large constant $C$, so $s \ge d$ and $q_i = \min\{1, s\widetilde{\tau}_i/d\} \ge \min\{1, C\epsilon^{-2}\log(d/\delta) \cdot \tau_i(A)\}$. We apply the matrix Chernoff bound for leverage score sampling with independent inclusions, Lemma 4 of Cohen et al. [2015b], as in item 3 of the proof of Lemma C.8. Therefore, $B := D^{\top}A$ satisfies $(1-\epsilon)A^{\top}A \preceq B^{\top}B \preceq (1+\epsilon)A^{\top}A$ with probability at least $1 - \delta/2$. By Step 2, $B$ has $O(s) = O(\epsilon^{-2}d\log(d/\delta))$ points, the expected time is $\widetilde{O}(n^{0.5}s^{0.5}d + d^{\omega}) = \widetilde{O}(\epsilon^{-1}n^{0.5}d^{1.5} + d^{\omega})$, and the number of queries is $\widetilde{O}(n^{0.5}s^{0.5}) = \widetilde{O}(\epsilon^{-1}n^{0.5}d^{0.5})$. □

### C.2 $\ell_p$ Sensitivity and Lewis Weights

To preserve $\ell_p$ subspace, one could sample according to $\ell_p$ sensitivity: let us define $\mathrm{cost}(a_i, x) = |a_i^\top x|^p$ for $p \in (0, \infty)$, then the $\ell_p$ sensitivity is

$$s_i(A, B) = \max_{x \in \mathbb{R}^d, Bx \neq 0} \frac{|a_i^\top x|^p}{\|Bx\|_p^p},$$

and a computationally efficient proxy for $\ell_p$ sensitivity is $\ell_p$ Lewis weights, defined as the unique nonnegative weight vector $w_A \in \mathbb{R}^n$ with

$$w_{A,i}^{2/p} = a_i^\top (A^\top W_A^{1-2/p} A)^{-1} a_i,$$

where $W_A \in \mathbb{R}^{n \times n}$ is the diagonal matrix of $w_A$. Naturally, we define our weights as

$$w_i(A, B) = (a_i^\top (B^\top W_B^{1-2/p} B)^{-1} a_i)^{p/2}.$$

To implement recursive sampling according to Lewis weights, we need a stronger notion of approximation for $\epsilon$-coreset, as beyond sensitivity, the weights might not be preserved by an $\epsilon$-coreset. We explicitly define the notion of an *$\epsilon$-approximator*, a weighted subset of points that preserves the Lewis quadratic form, and hence the weights. Note that an $\epsilon$-approximator is not necessarily an $\epsilon$-coreset.

**Definition C.13.** *Let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$ and $\epsilon \in (0, 1)$. We say a weighted subset $B$ of $A$ is an $\epsilon$-approximator of $A$ if*

$$(1 - \epsilon) \cdot A^\top W_A^{1-2/p} A \preceq B^\top W_B^{1-2/p} B \preceq (1 + \epsilon) \cdot A^\top W_A^{1-2/p} A,$$

*where $W_A$ and $W_B$ are the diagonal matrices of the $\ell_p$ Lewis weights of $A$ and of $B$.*

Note that the two-sided bound in Definition C.13 forces $\mathrm{rowspan}(B) = \mathrm{rowspan}(A)$. Moreover, by Lemma 2.6 of Woodruff and Yasuda [2025a], the pseudo-inverses satisfy $(1+\epsilon)^{-1}(A^\top W_A^{1-2/p} A)^\dagger \preceq (B^\top W_B^{1-2/p} B)^\dagger \preceq (1-\epsilon)^{-1}(A^\top W_A^{1-2/p} A)^\dagger$ on this row space. Hence, for any fixed $C$ and any $i$, we have $w_i(C, B)^{2/p} = (1 \pm 2\epsilon) \cdot w_i(C, A)^{2/p}$ when $\epsilon \leq 1/2$. Raising both sides to the power $p/2$ gives

$$w_i(C, B) = (1 \pm 2p\epsilon) \cdot w_i(C, A) \qquad \text{whenever } \epsilon \leq \min\{1/4, 1/p\},$$

where we use $(1 + 2\epsilon)^{p/2} \leq e^{p\epsilon} \leq 1 + 2p\epsilon$ for $p\epsilon \leq 1$ and $(1 - 2\epsilon)^{p/2} \geq 1 - 2p\epsilon$ for $\epsilon \leq 1/4$. Thus, an $\epsilon$-approximator preserves the weights up to a factor $1 \pm O(p\epsilon)$, and an $O(1/p)$-approximator preserves them up to a constant factor. For Lewis weights, the analogue of Lemma C.10 is the recursive algorithm of Lemma 6.2 of Cohen and Peng [2015]. It computes an approximate inverse Lewis quadratic form of a uniform half $\widehat{A}$ of $A$ recursively, and then samples the rows of $A$ with the relative weights given by this form.

For $p > 2$, the relative weights $(a_i^\top (\widehat{A}^\top W_{\widehat{A}}^{1-2/p} \widehat{A})^{-1} a_i)^{p/2}$ computed from the half $\widehat{A}$ are *not* entrywise overestimates of the Lewis weights of $A$. In fact, they fall below them on small random instances. The following lemma isolates one level of this recursion in the form our algorithm uses, and its proof follows the proof of Lemma 6.2 of Cohen and Peng [2015] step by step. In this lemma, the half is drawn by independent inclusions as in the chain of Algorithm 2, rather than as a subset of exactly $n/2$ rows. Moreover, the approximate inverse form is supplied by an approximator of the half, and the failure probability is made explicit. For $p < 2$, we also use a modification that Cohen and Peng [2015] note right after the proof of their Lemma 6.2. There, their Lemma 5.5 replaces their Lemma 5.6 for $p < 2$, and their Lemma 5.3 replaces their Lemma 5.4 for $p < 4$, so that for $p < 2$ no polynomial factor in $d$ multiplies the weights. Our proof uses Lemma 5.5 in Step 1 and Lemma 5.3 in Steps 3 and 4 in these ranges. As a consequence, for $p < 2$, the accuracy $\epsilon'$ of Step 2 needs no factor $1/\sqrt{d}$, and a level keeps $O_{p,\kappa}(\epsilon_1^{-2} d \log(d/\delta))$ rows in expectation. We write $M_A := A^\top W_A^{1-2/p} A$ for the Lewis quadratic form of $A$. For a rank-deficient matrix, the Lewis weights are taken on its row space and $M_A^\dagger$ replaces $M_A^{-1}$.

**Lemma C.14** (One level of the Lewis weight recursion)**.** *Let $p \in (0, \infty)$, let $A \in \mathbb{R}^{n \times d}$ have rows $a_1, \ldots, a_n$ (for a rank-deficient $A$, $d$ may be replaced by $\mathrm{rank}(A)$), and let $\epsilon_1, \delta \in (0, 1)$ and $\kappa \geq 1$. Let $\widehat{A} \subset_{1/2} A$ consist of the rows of $A$ included independently with probability $1/2$, let $C$ be a*

*$\frac{1}{4}$-approximator of $\widehat{A}$, and let $x_i := (a_i^\top M_C^\dagger a_i)^{p/2}$ for $a_i \in \mathrm{rowspan}(C)$ and $x_i := \infty$ otherwise. Let $\widehat{w}_1, \dots, \widehat{w}_n$ satisfy $\kappa^{-1} x_i \le \widehat{w}_i \le \kappa x_i$. Include each row $a_i$ independently with probability*

$$q_i = \min\{1, \beta \cdot \widehat{w}_i\}, \qquad \beta := c_{p,\kappa} \cdot \epsilon_1^{-2} \cdot d^{\theta_p} \cdot \log(d/\delta),$$

*scaled by $q_i^{-1/p}$, and let $A'$ be the resulting weighted subset. Here $\theta_p := 0$ for $p < 2$ and $\theta_p := p/2$ for $p \ge 2$. There is a constant $c_{p,\kappa}$ depending only on $p$ and $\kappa$ such that, for every outcome of $\widehat{A}$ and $C$, with probability at least $1 - \delta$ over the inclusions, $A'$ is an $\epsilon_1$-approximator of $A$, and moreover $\mathbb{E}[|A'|] = \mathbb{E}[\sum_{i=1}^n q_i] = O_{p,\kappa}(\epsilon_1^{-2} d^{\theta_p + 1} \log(d/\delta))$, where the expectation is over $\widehat{A}$ and the inclusions, for any $C$ and $\widehat{w}$ satisfying the hypotheses (which may depend on $\widehat{A}$).*

*Proof.* If $A$ is rank-deficient, we replace $A$ by $AV$, with $V \in \mathbb{R}^{d \times r}$ as in the paragraph after Lemma C.15. Every form $M$ built from rows of $A$ satisfies $M = V(V^\top M V)V^\top$ and $M^\dagger = V(V^\top M V)^\dagger V^\top$. Hence the Lewis weights, the quantities $x_i$, $y_i$, $v_i$ below, and the approximator relation between $A'$ and $A$ are unchanged. Moreover, $d$ enters the argument only as an upper bound on the rank of matrices with rows in $\mathrm{rowspan}(A)$, and we therefore assume that $A$ has full column rank. Since $C$ is a $\frac{1}{4}$-approximator of $\widehat{A}$, we have $\mathrm{rowspan}(C) = \mathrm{rowspan}(\widehat{A})$, and by Lemma 2.6 of Woodruff and Yasuda [2025a], $\frac{4}{5} M_{\widehat{A}}^\dagger \preceq M_C^\dagger \preceq \frac{4}{3} M_{\widehat{A}}^\dagger$ on this row space. Let $y_i := (a_i^\top M_{\widehat{A}}^\dagger a_i)^{p/2}$, and $y_i = \infty$ for $a_i \notin \mathrm{rowspan}(\widehat{A})$. Then $\kappa_1^{-1} y_i \le \widehat{w}_i \le \kappa_1 y_i$ for $\kappa_1 := \kappa(4/3)^{p/2}$. We write $w_i$ for $w_{A,i}$ throughout, and all inverses of forms of $\widehat{A}$ and of its extensions by one row are taken on $\mathrm{rowspan}(\widehat{A})$.

*Step 1: the relative weights dominate the Lewis weights of $A$ up to a factor $d^{(p/2-1)\vee 0}$.* If $p \le 2$, the monotonicity of Lewis weights in Lemma 5.5 of Cohen and Peng [2015] gives $M_A \succeq M_{\widehat{A}}$, hence $y_i \ge w_i$ for every $i$. If $p > 2$ and $\widehat{A}$ has full column rank, Lemma 5.6 of Cohen and Peng [2015] gives $M_{\widehat{A}} \preceq d^{1-2/p} M_A$. For $p > 2$ in general, Lemmas 2.3 and 2.4 of Woodruff and Yasuda [2025a] give the same bound by using $\|w_{\widehat{A}}\|_1 = \mathrm{rank}(\widehat{A}) \le d$ and $\widehat{A} \subseteq A$, hence $y_i \ge d^{-(p/2-1)} w_i$. In both cases, $y_i \ge d^{-((p/2-1)\vee 0)} w_i$. Since $d^{\theta_p} \cdot d^{-((p/2-1)\vee 0)}$ equals 1 for $p < 2$ and $d$ for $p \ge 2$, we obtain

$$q_i \ge \min\{1, \gamma \cdot w_i\}, \qquad \gamma := c_{p,\kappa} \kappa_1^{-1} \cdot \epsilon_1^{-2} \cdot d^{\mathbf{1}[p \ge 2]} \cdot \log(d/\delta).$$

Rows outside $\mathrm{rowspan}(\widehat{A})$ have $q_i = 1$.

*Step 2: the carried weights are almost Lewis weights of $A'$.* As in Cohen and Peng [2015], we give the sampled copy $a'_j = q_i^{-1/p} a_i$ of row $i$ the weight $w'_j := w_i / q_i$. Then $A'^\top W'^{1-2/p} A'$ is the Gram matrix of the rows of $W_A^{1/2-1/p} A$ included independently with probability $q_i$ and rescaled by $q_i^{-1/2}$, and the leverage scores of $W_A^{1/2-1/p} A$ are the Lewis weights $w_i$. Let $\epsilon' := \epsilon_1 / K_p$ if $p < 2$ and $\epsilon' := \epsilon_1 / (K_p \sqrt{d})$ if $p \ge 2$, for a constant $K_p$ depending only on $p$, which is chosen in Step 3 and accommodated by $c_{p,\kappa}$. In both cases, $\epsilon'^{-2} = K_p^2 \epsilon_1^{-2} d^{\mathbf{1}[p \ge 2]}$, and hence $\gamma = \Omega(\epsilon'^{-2} \log(d/\delta))$. Using the matrix Chernoff bound for independent inclusion in Lemma 4 of Cohen et al. [2015b], as in the proof of Lemma C.8, we have $(1 - \epsilon') \cdot M_A \preceq A'^\top W'^{1-2/p} A' \preceq (1 + \epsilon') \cdot M_A$ with probability at least $1 - \delta$. Hence $w'$ is a vector of $(1 + 2\epsilon')$-almost Lewis weights of $A'$ in the sense of Definition 5.1 of Cohen and Peng [2015].

*Step 3: stability transfers the guarantee to the true Lewis weights of $A'$.* Let $\overline{w}'$ be the vector of Lewis weights of $A'$ and let $\eta := \epsilon_1 / (4(1 \vee |1 - 2/p|))$. For $p < 4$, Lemma 5.3 of Cohen and Peng [2015] gives $\overline{w}'_j = (1 + 2\epsilon')^{\pm c_p} w'_j$ with $c_p = (p/2)/(1 - |p/2 - 1|)$. For $p \ge 2$, Lemma 5.4 of Cohen and Peng [2015] gives $\overline{w}'_j = (1 + 2\epsilon')^{\pm K_0 f_0(p) \sqrt{d}} w'_j$ for an absolute constant $K_0$ and a function $f_0$ of $p$ alone. By choosing $K_p$ large enough as a function of $p$, we have $\overline{w}'_j = (1 \pm \eta) w'_j$ for all $j$, using Lemma 5.3 when $p < 4$ and Lemma 5.4 when $p \ge 4$, and we also have $\epsilon' \le \epsilon_1 / 4$. For $p < 2$, the accuracy $\epsilon' = \epsilon_1 / K_p$ does not depend on $d$, and this suffices because $c_p = 1$. Indeed, $K_p \ge 8(1 \vee |1 - 2/p|)$ gives $2\epsilon' \le \eta$ and $\epsilon' \le \epsilon_1 / 8$, so both values $(1 + 2\epsilon')^{\pm 1}$ lie in $[1 - \eta, 1 + \eta]$, that is, $\overline{w}'_j = (1 \pm \eta) w'_j$. Hence $(\overline{w}'_j)^{1-2/p} = (1 \pm \epsilon_1/2) (w'_j)^{1-2/p}$, and therefore

$$M_{A'} = A'^\top \overline{W}'^{1-2/p} A' = (1 \pm \epsilon_1/2) \cdot A'^\top W'^{1-2/p} A' = (1 \pm \epsilon_1/2)(1 \pm \epsilon_1/4) \cdot M_A,$$

which gives $(1-\epsilon_1)M_A \preceq M_{A'} \preceq (1+\epsilon_1)M_A$. Thus, $A'$ is an $\epsilon_1$-approximator of $A$.

*Step 4: expected size.* For $i \in [n]$, let $\widehat{A}_i$ be $\widehat{A}$ with the row $a_i$ appended if $a_i \notin \widehat{A}$, and $\widehat{A}_i = \widehat{A}$ otherwise. Let $v_i := w_{\widehat{A}_i,i}$ be the Lewis weight of $a_i$ in $\widehat{A}_i$. As in the proof of Theorem C.3, we condition on $|\widehat{A}| = m$, and given this event, $\widehat{A}$ is a uniformly random subset of $m$ rows. As in Part 2 of the proof of Lemma 6.2 of Cohen and Peng [2015], consider the random process that draws such an $\widehat{A}$ and then a uniformly random $i \notin \widehat{A}$. It is the same as drawing a uniformly random subset of $m+1$ rows and then a uniformly random row in it. Since the Lewis weights of any matrix sum to its rank, which is at most $d$, this equivalence gives $\mathbb{E}[\sum_{i\notin\widehat{A}} v_i \mid |\widehat{A}| = m] \le (n-m)\cdot\frac{d}{m+1} \le \frac{n+1}{m+1}\cdot d$. We then average over $m$ using $\mathbb{E}[\frac{n+1}{|\widehat{A}|+1}] = 2 - 2^{-n} \le 2$, which is computed in the proof of Theorem C.3, and obtain $\mathbb{E}[\sum_{i\notin\widehat{A}} v_i] \le 2d$. Since $\sum_{i\in\widehat{A}} v_i = \sum_{i\in\widehat{A}} w_{\widehat{A},i} \le d$ for every outcome, we have $\mathbb{E}[\sum_{i=1}^n v_i] \le 3d$. It remains to show that $q_i \le O_{p,\kappa}(\beta)\cdot v_i$ for every $i$, pointwise for every $C$ and $\widehat{w}$ satisfying the hypotheses. If $a_i \in \widehat{A}$, then $y_i = v_i$ and $q_i \le \beta\kappa_1 v_i$. If $a_i \notin \operatorname{rowspan}(\widehat{A})$, then $v_i = 1$ and $q_i = 1 = v_i$. The last row of $W^{1/2-1/p}\widehat{A}_i$ lies outside the span of the other rows for every positive diagonal $W$, so its leverage score is 1. Otherwise, let $\widetilde{v}$ be the Lewis weights of $\widehat{A}_i$ and let $N := \sum_{j\in\widehat{A}} \widetilde{v}_j^{1-2/p} a_j a_j^\top$. Then $M_{\widehat{A}_i} = N + v_i^{1-2/p} a_i a_i^\top$ and $v_i^{1-2/p}\cdot a_i^\top M_{\widehat{A}_i}^{-1} a_i = v_i^{1-2/p} v_i^{2/p} = v_i$. By the Sherman–Morrison formula and the Cauchy–Schwarz inequality, we have for every $j \in \widehat{A}$,

$$a_j^\top M_{\widehat{A}_i}^{-1} a_j \le a_j^\top N^{-1} a_j = a_j^\top M_{\widehat{A}_i}^{-1} a_j + \frac{v_i^{1-2/p}(a_j^\top M_{\widehat{A}_i}^{-1} a_i)^2}{1-v_i} \le \frac{a_j^\top M_{\widehat{A}_i}^{-1} a_j}{1-v_i},$$

which means that $\widetilde{v}$ restricted to $\widehat{A}$ is a vector of $(1-v_i)^{-1}$-almost Lewis weights of $\widehat{A}$. Likewise, $a_i^\top N^{-1} a_i = v_i^{2/p}/(1-v_i)$. Let $\eta_p := 1/2$ if $p < 4$ and $\eta_p := \min\{1/2, 1/(C' f_0(p)\sqrt{d})\}$ if $p \ge 4$, for an absolute constant $C'$. Then $1-\eta_p \ge 1/2$ in both cases. If $v_i \le \eta_p$, then the stability lemmas, Lemma 5.3 of Cohen and Peng [2015] for $p < 4$ and Lemma 5.4 for $p \ge 4$, give $\widetilde{v}_j = \Theta_p(1)\cdot w_{\widehat{A},j}$ for all $j \in \widehat{A}$. In that case, $N = \Theta_p(1)\cdot M_{\widehat{A}}$ in the Loewner order, so $a_i^\top M_{\widehat{A}}^{-1} a_i \le O_p(1)\cdot a_i^\top N^{-1} a_i \le O_p(1)\cdot v_i^{2/p}$, and hence $y_i \le O_p(1)\cdot v_i$ and $q_i \le \beta\kappa_1 \cdot O_p(1)\cdot v_i$. If $v_i > \eta_p$, then $q_i \le 1 < v_i/\eta_p$. For $p < 4$, we have $v_i/\eta_p = 2v_i \le O_p(\beta)\cdot v_i$, and for $p \ge 4$, we have $v_i/\eta_p \le O_p(\sqrt{d})\cdot v_i \le O_p(\beta)\cdot v_i$ since $\theta_p = p/2 \ge 2$. Thus, in all cases $q_i \le O_{p,\kappa}(\beta)\cdot v_i$, and by taking expectations, we have

$$\mathbb{E}\Big[\sum_{i=1}^n q_i\Big] \le O_{p,\kappa}(\beta)\cdot\mathbb{E}\Big[\sum_{i=1}^n v_i\Big] \le O_{p,\kappa}(\beta d) = O_{p,\kappa}(\epsilon_1^{-2} d^{\theta_p+1}\log(d/\delta)). \qquad \square$$

We also need the following result, due to Cohen and Peng [2015] for $p \in (0,4)$ and to Fazel et al. [2022] for $p \ge 4$, that approximates $\ell_p$ Lewis weights in nearly exact leverage score time:

**Lemma C.15** (Lemma 2.4 and Corollary 3.3 of Cohen and Peng [2015] and Theorem 1 of Fazel et al. [2022])**.** *Let $A = \{a_1, \dots, a_n\} \subset \mathbb{R}^d$ with $A \in \mathbb{R}^{n\times d}$ of full column rank, $p \in (0,\infty)$ and $\epsilon \in (0,1)$. Then, there exists a deterministic algorithm that outputs a vector $\widetilde{w}_A \in \mathbb{R}^n$ such that for any $i \in [n]$, $\widetilde{w}_{A,i} = (1\pm\epsilon)\cdot w_{A,i}$. Moreover, the runtime of this algorithm is*

$$O_p(nd^{\omega-1}\log(np/\epsilon)).$$

The approximation notion in the lemma is entrywise closeness to the true Lewis weights, which is the notion (1.5) of Fazel et al. [2022]. For $p \in (0,4)$, the algorithm is the fixed-point iteration $w_i \leftarrow (a_i^\top (A^\top W^{1-2/p} A)^{-1} a_i)^{p/2}$ of Cohen and Peng [2015] with exact leverage scores, which needs $O_p(\log(\log n/\epsilon))$ steps by Lemma 3.5, Lemma 3.2, and Corollary 3.3 of that paper. For $p \ge 4$, and in fact for all $p > 2$ by footnote 5 of Fazel et al. [2022], Theorem 1 of Fazel et al. [2022] performs $O(p^3\log(np/\epsilon))$ leverage score computations of matrices $DA$ with $D$ diagonal. In both cases, each leverage score computation is for an $n \times d$ matrix and takes $O(nd^{\omega-1})$ time. Both sources assume full column rank, so for a rank-deficient $A$ we apply the lemma to $AV$, where $V \in \mathbb{R}^{d\times r}$ is an orthonormal basis of $\operatorname{rowspan}(A)$ and $r = \operatorname{rank}(A)$. This matrix has the same Lewis weights as $A$, since the weights are defined through $a_i^\top (A^\top W^{1-2/p} A)^\dagger a_i$ and $(A^\top W^{1-2/p} A)^\dagger = V(V^\top A^\top W^{1-2/p} A V)^{-1} V^\top$ by Lemma 2.5 of Woodruff and Yasuda [2025a].

**Algorithm 5** Classical oracle for Lewis weights.

**data structure** LEWISWEIGHTS
**members**
  $A \in \mathbb{R}^{n \times d}$
  $C \in \mathbb{R}^{s \times d}$
  $M \in \mathbb{R}^{O_p(\log(n/\delta)) \times d}$
  $N \in \mathbb{R}^{O(\log n) \times d}$
**end members**

**procedure** PREPROCESS($A \in \mathbb{R}^{n \times d}, C \in \mathbb{R}^{s \times d}, \delta \in (0,1)$)
  $\gamma \leftarrow \min\{1/8, 1/(8p)\}$, $m \leftarrow c\gamma^{-2}\log(n/\delta)$ for an absolute constant $c$
  Compute an orthonormal basis $V_C \in \mathbb{R}^{d \times r}$ of $\mathrm{rowspan}(C)$, $r = \mathrm{rank}(C)$
  Generate $\widetilde{W}_C$ via Lemma C.15 on $CV_C$ with accuracy $\gamma$ ▷ $\widetilde{W}_C \in \mathbb{R}^{s \times s}$
  Compute the compact SVD of $\widetilde{W}_C^{1/2-1/p} C$: $\widetilde{W}_C^{1/2-1/p} C = U \Sigma V^\top$ ▷ $\Sigma \in \mathbb{R}^{r \times r}$ invertible, $V \in \mathbb{R}^{d \times r}$, $\mathrm{span}(V) = \mathrm{rowspan}(C)$
  Let $G \in \mathbb{R}^{m \times r}$ and $G' \in \mathbb{R}^{c \log n \times d}$ be random Gaussian matrices
  $M \leftarrow G\Sigma^{-1}V^\top$ ▷ $M \in \mathbb{R}^{m \times d}$
  $N \leftarrow G'(I - VV^\top)$ ▷ sketch of the orthogonal complement of $\mathrm{rowspan}(C)$
**end procedure**

**procedure** QUERY($i \in [n]$)
  **if** $Na_i \neq 0$ **then return** $\infty$ ▷ $a_i \notin \mathrm{rowspan}(C)$, so $q_i = 1$
  **return** $\|Ma_i\|_2^p$
**end procedure**
**end data structure**

**Lemma C.16.** *Let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$, $p \in (0, \infty)$, $\delta \in (0,1)$, $C \in \mathbb{R}^{s \times d}$, and define $w(A, C)$ as*

$$w_i(A,C)^{2/p} = \begin{cases} a_i^\top (C^\top W_C^{1-2/p} C)^\dagger a_i, & \text{if } a_i \in \mathrm{rowspan}(C), \\ \infty, & \text{otherwise}, \end{cases}$$

*where $W_C$ is the diagonal matrix of the $\ell_p$ Lewis weights of $C$, taken on its row space. Then $\sum_{i=1}^n w_i(A,A) = \mathrm{rank}(A) \leq d$, so $\mathrm{sum}(w) = d$, and there exists a randomized algorithm (Algorithm 5) that implements* PREPROCESS *and* QUERY *procedures such that, with probability at least $1 - \delta$ over the randomness of* PREPROCESS*, every query $i \in [n]$ returns a value $\widehat{w}_i$ with $\frac{1}{2} \cdot w_i(A,C) \leq \widehat{w}_i \leq 2 \cdot w_i(A,C)$ (and $\widehat{w}_i = \infty$ exactly when $w_i(A,C) = \infty$), with*

- $\mathcal{T}_{\mathrm{prep}}(s,d) = \widetilde{O}_p(sd^{\omega-1})$.
- $\mathcal{T}_{\mathrm{query}}(s,d) = \widetilde{O}_p(d)$.

*Proof.*
- Consistent total weights: observe that we can alternatively define $w_A$ as $w_{A,i} = \tau_i(W_A^{1/2-1/p} A)$, i.e., it is the leverage score of $W_A^{1/2-1/p} A$. Since the sum of the leverage scores is the rank, which is at most $d$, we have $\mathrm{sum}(w) = d$.
- Approximator preserves weights: this follows from Definition C.13, as shown in the paragraph after it. Therefore, to preserve the weights up to a constant factor, we work with $O(1/p)$-approximators, and in the end we do a final resampling using the approximated Lewis weights.

To analyze Algorithm 5, we first consider a variant where the Johnson–Lindenstrauss transformation is not applied. Let $V_C$ be an orthonormal basis of $\mathrm{rowspan}(C)$. As shown in the paragraph after Lemma C.15, the matrix $CV_C$ has full column rank and the same Lewis weights as $C$, so Lemma C.15 applies to it. We compute $\widetilde{W}_C$ using Lemma C.15 with accuracy $\gamma = \min\{1/8, 1/(8p)\}$, which is a $1 \pm \gamma$ entrywise approximation to $W_C$. Then we know that $\widetilde{W}_C^{1-2/p}$ is a $(1 \pm 2\gamma)^{|1-2/p|}$ entrywise approximation to $W_C^{1-2/p}$, and this approximation guarantee propagates to $C^\top \widetilde{W}_C^{1-2/p} C$

and, by Lemma 2.6 of Woodruff and Yasuda [2025a], to its pseudo-inverse on $\mathrm{rowspan}(C)$. So far, we have established that for any $a_i \in \mathrm{rowspan}(C)$, $a_i^\top (C^\top \widetilde{W}_C^{1-2/p} C)^\dagger a_i = (1 \pm 2\gamma)^{|1-2/p|} \cdot a_i^\top (C^\top W_C^{1-2/p} C)^\dagger a_i$. Our final output is the $(p/2)$-th power of the quantity, hence the approximate weight is a $(1 \pm 2\gamma)^{|p/2-1|}$ approximation to the true quantity $w_i(A, C)$. Rows $a_i \notin \mathrm{rowspan}(C)$ have $w_i(A, C) = \infty$, and the test $Na_i \neq 0$ in QUERY detects exactly these rows with probability 1. This is the same exact zero test as in the proof of Lemma C.8, and $V$ can be taken equal to $V_C$.

When applying the Johnson–Lindenstrauss transformation, we are effectively approximating $a_i^\top (C^\top \widetilde{W}_C^{1-2/p} C)^\dagger a_i = \|\Sigma^{-1} V^\top a_i\|_2^2$ within a factor $1 \pm \gamma$, simultaneously for all $n$ rows and with probability at least $1 - \delta$, by a Gaussian $G$ with $m = O(\gamma^{-2} \log(n/\delta)) = O_p(\log(n/\delta))$ rows [Spielman and Srivastava, 2011]. Combining the two approximations gives $\widehat{w}_i = \|Ma_i\|_2^p = \big((1 \pm 2\gamma)^{|1-2/p|}(1 \pm \gamma)\big)^{p/2} \cdot w_i(A, C)$, and for $\gamma = \min\{1/8, 1/(8p)\}$, the multiplicative factor lies in $[1/2, 2]$ for every $p > 0$. For $p < 2$ the factor lies in $[1 - 2\gamma, 1 + 2\gamma]$, and for $p \geq 2$ it lies in $[e^{-3\gamma p}, e^{3\gamma p/2}] \subseteq [e^{-3/8}, e^{3/16}]$. Let us analyze the runtime.

- PREPROCESS: computing $V_C$ takes $O(sd^{\omega-1})$ time. To compute $\widetilde{W}_C$, we set the accuracy in Lemma C.15 to $\gamma = \Theta(1/(1 \vee p))$, and it runs in time $\widetilde{O}_p(sd^{\omega-1})$. Computing the compact SVD takes $O(sd^{\omega-1})$ time, and applying the random Gaussian matrices $G$ and $G'$ takes $\widetilde{O}_p(sd)$ time.
- QUERY: note that $M \in \mathbb{R}^{\widetilde{O}_p(1) \times d}$ and $N \in \mathbb{R}^{O(\log n) \times d}$, hence answering one query takes time $\widetilde{O}_p(d)$.

This completes the proof. □

Lemmas C.14 and C.16 give an approach to compute an $O(1/p)$-approximator for $A$, but our ultimate goal is to compute an $\epsilon$-coreset for $A$, which has a different objective. The following result states that sampling according to the appropriate scaling of overestimates of Lewis weights indeed yields an $\epsilon$-coreset:

**Lemma C.17** (Theorems 1.3 and A.2 of Woodruff and Yasuda [2025a])**.** *Let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$, $\epsilon, \delta \in (0, 1)$ and $p \in (0, \infty)$ and let $u \in \mathbb{R}^n$ be an overestimate of $w_A$ with $\|u\|_1 \leq O(d)$. Consider the sampling scheme where each point is sampled with probability $q_i = \min\{1, \alpha \cdot u_i\}$ where*

- $\alpha = \epsilon^{-2}(\log^3 d + \log(1/\delta))$ *for* $p \in (0, 1)$.
- $\alpha = \epsilon^{-2} \log(n/\delta)$ *for* $p = 1$.
- $\alpha = \epsilon^{-2}(\log^2 d \log n + \log(1/\delta))$ *for* $p \in (1, 2)$.
- $\alpha = \epsilon^{-2} d^{p/2-1}(\log^2 d \log n + \log(1/\delta))$ *for* $p \geq 2$.

*Let $S \in \mathbb{R}^{n \times n}$ be the diagonal matrix with $S_{ii} = q_i^{-1/p}$ for the sampled points and $S_{ii} = 0$ otherwise. Then, with probability at least $1 - \delta$, $SA$ is an $\epsilon$-coreset of $A$, and the number of samples has expectation $\sum_i q_i \leq \alpha \cdot \|u\|_1$ and is $O(\alpha \cdot \|u\|_1)$ with probability at least $1 - \delta$.*

Theorem 1.3 of Woodruff and Yasuda [2025a] is stated for $p > 2$ and for *one-sided* Lewis weights, which Definition 2.2 there defines as vectors $u$ with $u_i \geq (a_i^\top (A^\top U^{1-2/p} A)^{-1} a_i)^{p/2}$ for all $i$, where $U = \mathrm{diag}(u)$. When $p > 2$, every entrywise overestimate $u$ of $w_A$ is one-sided, since $1 - 2/p > 0$ gives $U^{1-2/p} \succeq W_A^{1-2/p}$, and hence $a_i^\top (A^\top U^{1-2/p} A)^{-1} a_i \leq a_i^\top (A^\top W_A^{1-2/p} A)^{-1} a_i = w_{A,i}^{2/p} \leq u_i^{2/p}$. For $p < 2$, Theorem A.2 of Woodruff and Yasuda [2025a] requires only that each point be sampled with probability at least $\min\{1, \alpha \cdot w_{A,i}\}$, and $q_i = \min\{1, \alpha \cdot u_i\}$ satisfies this for any overestimate $u$. For $p = 2$, the statement is standard leverage score sampling. The number of samples is a sum of independent Bernoulli variables with mean $\sum_i q_i \leq \alpha \|u\|_1$. Since $\alpha \geq \log(1/\delta)$ in every case, by a Chernoff bound, the number of samples is at most $2\alpha\|u\|_1 + O(\log(1/\delta)) = O(\alpha\|u\|_1)$ with probability at least $1 - \delta$.

**Theorem C.18.** *Let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$, $\epsilon, \delta \in (0,1)$ and $p \in (0, \infty)$. There exists a quantum algorithm that with probability at least $1 - \delta$, constructs an $\epsilon$-coreset of $A$ with size $O_p(\alpha \cdot d)$, for $\alpha$ given in Lemma C.17. The runtimes for generating the coreset, in expectation conditioned on the success event of probability at least $1 - \delta$ (as in Theorem C.3), are*

- $\widetilde{O}_p(\epsilon^{-1} n^{0.5} d^{1.5} + d^{\omega})$ *for* $p \in (0, 2)$.
- $\widetilde{O}_p(d^{p/2+\omega}) + \widetilde{O}_p(n^{0.5} d^{p/4+1}(\epsilon^{-1} + d^{0.5}))$ *for* $p \geq 2$.

*The numbers of queries to the points in $A$ for generating the coreset, in the same expectation, are*

- $\widetilde{O}_p(\epsilon^{-1} n^{0.5} d^{0.5})$ *for* $p \in (0, 2)$.
- $\widetilde{O}_p(n^{0.5} d^{p/4}(\epsilon^{-1} + d^{0.5}))$ *for* $p \geq 2$.

*Proof.* Our strategy will be to first construct an $O(1/p)$-approximator of $A$, which in turn gives an $O(1)$-approximation to $w_A$, then we will sample according to these approximations, in conjunction with Lemma B.15. We assume that $A$ has full column rank. Otherwise, every statement below holds on $\mathrm{rowspan}(A)$ with $d$ replaced by $\mathrm{rank}(A)$. Note that the intermediate sets $A_t$ may be rank-deficient even when $A$ is not, but Lemma C.14 does not require full column rank and is applied to them with $d \geq \mathrm{rank}(A_t)$. Set $\epsilon_1 := \min\{1/4, 1/(4p)\}$, let $s'$ be the quantity chosen in Stage 1 below, and let $T := \lceil \log_2(n/s') \rceil = O(\log n)$.

- Stage 1: constructing an $\epsilon_1$-approximator, hence an $O(1/p)$-approximator, of $A$. We run the recursion of Algorithm 2 on the chain $A_0 \subset_{1/2} A_1 \subset_{1/2} \cdots \subset_{1/2} A_T = A$ with $C_0 = A_0$, using the oracle of Algorithm 5 with the same accuracy $\epsilon_1$ at every level, so $s = s'$. We replace the sampling step of every level by the rule of Lemma C.14 and use the relative weights without augmentation, since that lemma handles the rows outside $A_{t-1}$ directly and takes the place of the conditions of Assumption C.1. We use Theorem C.3 only to account for the time and the queries. At level $t$, we sample $A_t$ with

$$q_i = \min\{1, \beta \cdot \widehat{w}_i\}, \qquad \beta := c_{p,2} \cdot \epsilon_1^{-2} \cdot d^{\theta_p} \cdot \log(8Td/\delta),$$

  where $\widehat{w}_i$ is the output of $\textsc{Query}(i)$ after $\textsc{Preprocess}(A_t, C_{t-1}, \delta/(8T))$, $c_{p,2}$ is the constant of Lemma C.14 for $\kappa = 2$, and $\theta_p$ is as in that lemma, that is, $\theta_p = 0$ for $p < 2$ and $\theta_p = p/2$ for $p \geq 2$. We show by induction on $t$ that with probability at least $1 - \delta/4$, $C_t$ is an $\epsilon_1$-approximator of $A_t$ for every $t \leq T$. The base case is trivial since $C_0 = A_0$. Suppose $C_{t-1}$ is an $\epsilon_1$-approximator of $A_{t-1}$, the Bernoulli$(1/2)$ half of $A_t$. Since $\epsilon_1 \leq 1/4$, Lemma C.16 gives $\widehat{w}_i = 2^{\pm 1} \cdot w_i(A_t, C_{t-1})$ for all $a_i \in A_t$ with probability at least $1 - \delta/(8T)$, and Lemma C.14 with $\kappa = 2$ and failure probability $\delta/(8T)$ shows that $C_t$ is an $\epsilon_1$-approximator of $A_t$. Thus, the $2T$ events have total failure probability $\delta/4$. For $t \in [T]$, let $\mathcal{H}_t$ be the event that $C_{t-1}$ is an $\epsilon_1$-approximator of $A_{t-1}$ and the oracle guarantee of level $t$ holds. On $\mathcal{H}_t$, the hypotheses of Lemma C.14 are met with $\kappa = 2$ and $\widehat{A} = A_{t-1} \subset_{1/2} A_t$. In this case, Step 4 of the proof of that lemma bounds $\sum_{i \in A_t} q_i$ pointwise by $O_{p,2}(\beta) \sum_{i \in A_t} v_i$, where $v_i$ depends only on $A_t$ and $A_{t-1}$ and $\mathbb{E}[\sum_{i \in A_t} v_i \mid A_t] \leq 3d$. Therefore, when the sampler call of level $t$ returns an exact sample, an event included in $\mathcal{S}$ below, we have $\mathbb{E}[|C_t| \, \mathbf{1}_{\mathcal{H}_t}] = \mathbb{E}[\sum_{i \in A_t} q_i \, \mathbf{1}_{\mathcal{H}_t}] \leq s'$ for

$$s' := O_p(\epsilon_1^{-2} d^{\theta_p + 1} \log(Td/\delta)) = \widetilde{O}_p(d^{\theta_p + 1}),$$

  which is $\widetilde{O}_p(d)$ for $p < 2$, and $\mathbb{E}\,|A_0| = n 2^{-T} \leq s'$ by the choice of $T$. Let $\mathcal{S}$ be the intersection of the $2T$ events of this stage, the success of the $T + 2$ calls of the sampler of Lemma B.15 in the two stages, and the three events of Stage 2. The sampler is called for $A_0$, for the $T$ levels, and for the final sample, each time with failure probability $\delta/(8(T+2))$, so these calls fail with probability at most $\delta/8$ in total. By the last item below, the event $\mathcal{S}$ has probability at least $1 - 6\delta/8 \geq 1/4$, and it is contained in every $\mathcal{H}_t$. Consequently, we have $\mathbb{E}[|C_t| \mid \mathcal{S}] \leq s'/\Pr[\mathcal{S}] \leq 4s'$ and $\mathbb{E}[\sum_{i \in A_t} q_i \mid \mathcal{S}] \leq 4s'$ for every $t \leq T$. As in Theorem C.3, we condition all expectations in the running time analysis on $\mathcal{S}$. Each level prepares one oracle in $\mathcal{T}_{\mathrm{prep}}(|C_{t-1}|, d) = \widetilde{O}_p(|C_{t-1}| d^{\omega-1})$ time, which is linear in $|C_{t-1}|$

and hence has conditional expectation $\widetilde{O}_p(s'd^{\omega-1})$. It then samples using Lemma B.15 in $\widetilde{O}(\sqrt{n\sum_{i\in A_t} q_i})\cdot\mathcal{T}_{\text{query}}$ time. Therefore, by Jensen's inequality, the expected time of Stage 1 conditioned on $\mathcal{S}$ is

$$\widetilde{O}_p(d^{\theta_p+\omega})+\widetilde{O}_p(n^{0.5}d^{\theta_p/2+1.5}).$$

We let $\widetilde{B}:=C_T$ denote the resulting approximator. Note that the expected size of $\widetilde{B}$ conditioned on $\mathcal{S}$ is $\widetilde{O}_p(d^{\theta_p+1})$. In particular, for $p<2$, Stage 1 takes $\widetilde{O}_p(d^\omega+n^{0.5}d^{1.5})$ expected time, and $\widetilde{B}$ has $\widetilde{O}_p(d)$ rows in expectation.

- Stage 2: constructing an $\epsilon$-coreset of $A$. Observe that $\widetilde{B}$ gives an $O(1)$-approximation to $w_A$. Since $\widetilde{B}$ is an $\epsilon_1$-approximator of $A$ with $\epsilon_1\le\min\{1/4,1/p\}$, by the consequence of Definition C.13, for any $a_i$ we have

$$\begin{aligned}(a_i^\top(\widetilde{B}^\top W_{\widetilde{B}}^{1-2/p}\widetilde{B})^{-1}a_i)^{p/2}&=(1\pm 2p\epsilon_1)\cdot(a_i^\top(A^\top W_A^{1-2/p}A)^{-1}a_i)^{p/2}\\&=(1\pm 1/2)\cdot w_{A,i},\end{aligned}$$

and after appropriately rescaling this yields the desired oversampling vector $u$. Concretely, we run PREPROCESS$(A,\widetilde{B},\delta/8)$ of Algorithm 5 and set $u_i:=4\cdot\widehat{w}_i$, where $\widehat{w}_i=2^{\pm1}\cdot w_i(A,\widetilde{B})$ is the output of QUERY$(i)$. Then we have $u_i\ge 4\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot w_{A,i}=w_{A,i}$ for all $i$, and

$$\begin{aligned}\|u\|_1=4\sum_{i=1}^n\widehat{w}_i&\le 8\sum_{i=1}^n(a_i^\top(\widetilde{B}^\top W_{\widetilde{B}}^{1-2/p}\widetilde{B})^{-1}a_i)^{p/2}\\&\le 12\cdot\sum_{i=1}^n(a_i^\top(A^\top W_A^{1-2/p}A)^{-1}a_i)^{p/2}\\&=12d,\end{aligned}$$

and we could invoke Lemma C.17 with $\delta/8$ in place of $\delta$ to generate the desired $\epsilon$-coreset. Since $\widetilde{B}$ has expected size $\widetilde{O}_p(d^{\theta_p+1})$ conditioned on $\mathcal{S}$ and the cost of PREPROCESS is linear in $|\widetilde{B}|$, generating $\widetilde{W}_{\widetilde{B}}$ takes $\widetilde{O}_p(d^{\theta_p+\omega})$ expected time. Computing the SVD and applying the Gaussian matrices takes $\widetilde{O}_p(d^{\theta_p+\omega})$ expected time as well. Each query takes $\widetilde{O}_p(d)$ time. Moreover, $\alpha\|u\|_1=\widetilde{O}_p(\epsilon^{-2}d^{p/2\vee1})$ on $\mathcal{S}$, so sampling according to $q_i=\min\{1,\alpha\cdot u_i\}$ with Lemma B.15 takes $\widetilde{O}_p(\sqrt{n\alpha\|u\|_1})\cdot\widetilde{O}_p(d)=\widetilde{O}_p(\epsilon^{-1}n^{0.5}d^{(p/2\vee1)/2+1})$ time. All in all, we obtain the following runtime for generating the $\epsilon$-coreset, where in each line the first term is the preprocessing cost of the two stages, the second term is the sampling cost of Stage 2, and the third term is the sampling cost of Stage 1:

  - For $p\in(0,2)$, it takes time $\widetilde{O}_p(d^\omega+\epsilon^{-1}n^{0.5}d^{1.5}+n^{0.5}d^{1.5})$, which is $\widetilde{O}_p(\epsilon^{-1}n^{0.5}d^{1.5}+d^\omega)$ since $\epsilon<1$.
  - For $p\ge2$, it takes time $\widetilde{O}_p(d^{p/2+\omega}+\epsilon^{-1}n^{0.5}d^{p/4+1}+n^{0.5}d^{p/4+1.5})$.

  The query complexity follows from Theorem C.3 and Lemma B.15. Stage 1 makes $\widetilde{O}_p(\sqrt{ns'})=\widetilde{O}_p(n^{0.5}d^{\theta_p/2+0.5})$ queries to the points of $A$, and Stage 2 makes $\widetilde{O}_p(\sqrt{n\alpha\|u\|_1})=\widetilde{O}_p(\epsilon^{-1}n^{0.5}d^{(p/2\vee1)/2})$ queries. For $p\in(0,2)$, both counts are at most $\widetilde{O}_p(\epsilon^{-1}n^{0.5}d^{0.5})$, since $\theta_p=0$ and $\epsilon<1$.

- Failure probability and size. The three events of Stage 2 in $\mathcal{S}$ are the oracle guarantee, the coreset guarantee of Lemma C.17, and its size guarantee, each with failure probability $\delta/8$. Together with the failure probabilities of the level events and the sampler calls computed in Stage 1, a union bound shows that $\mathcal{S}$ has probability at least $1-\delta/4-\delta/8-3\delta/8\ge1-\delta$. On $\mathcal{S}$, the output is an $\epsilon$-coreset of $A$ of size $O(\alpha\|u\|_1)=O_p(\alpha d)$. This is because $\|u\|_1\le12d$ holds once $\widetilde{B}$ is an approximator and the Stage 2 oracle guarantee holds, and both are part of $\mathcal{S}$.

This concludes the proof. □

**Remark C.19** (Row sparsity)**.** *Suppose that every row of $A$ has at most $r \le d$ nonzero entries (this $r$ is not the rank of $C$ in the oracles of this section) and that a row query returns these entries with their positions. Then the procedure* QUERY *of Algorithms 3, 5, and 6 takes $\widetilde{O}_p(r)$ time instead of $\widetilde{O}_p(d)$, since it uses $a_i$ only through its inner products with the $O_p(\log(n/\delta))$ rows of $M$ and $N$, which are held in QRAM (for rational inputs, the fingerprint of the zero test in the proof of Lemma C.8 can be taken as one more inner product, with a random combination of the basis of the orthogonal complement of the row space of $C$). The other work per row, namely reading the points of $A_0$, of the samples $C_t$, and of the output, and computing each $q_i$ and weight from the value returned by* QUERY*, takes $O(r)$ time, and the membership tests of the chain read no row. Therefore, the query counts of Theorems C.12, C.18, and C.25 do not change, and since each of their running times is $\widetilde{O}_p(Qd)$ for the query count $Q$ plus additive preprocessing terms, such as $d^\omega$, $d^{p/2+\omega}$, and $dk^{\omega-1}$, the running times become $\widetilde{O}_p(Qr)$ plus the same additive terms. For example, Theorem C.18 for $p \in (0,2)$ and Theorem C.12 for $p = 2$ give time $\widetilde{O}_p(\epsilon^{-1}n^{0.5}d^{0.5}r + d^\omega)$, which matches the leading term $\widetilde{O}(r\sqrt{nd}/\epsilon)$ of Corollary 11 of Liu and Ji [2026] (arXiv v1). This term comes from computing their sparsifier, which takes $\widetilde{O}(r\sqrt{nd}/\epsilon + d^\omega + dr^2)$ time by their Theorem 10 applied to $\ell_p$ losses (their Section 4.2), and their Corollary 11 adds $\mathrm{poly}(d, 1/\epsilon)$ time to solve the regression on the sparsifier. Table 1 compares our time with this sparsifier time. For simplicity, we state all our bounds for dense points, that is, $r = d$.*

### C.3 $k$-Subspace Sensitivity and Ridge Leverage Score

Let $\mathcal{F}_k$ be the set of all $k$-dimensional subspaces in $\mathbb{R}^d$. We can define the cost with respect to a subspace by identifying $X = \mathcal{F}_k$ and $\mathrm{cost} : \mathbb{R}^d \times \mathcal{F}_k \to \mathbb{R}_{\ge 0}$ as

$$\mathrm{cost}(a_i, x) = \|a_i^\top (I - P_x)\|_2^2,$$

where $P_x$ is the orthogonal projection onto $x$. Then, the $k$-subspace sensitivity is

$$s_i(A, B) = \max_{x \in X} \frac{\|a_i^\top (I - P_x)\|_2^2}{\|B(I - P_x)\|_F^2}.$$

Similar to $\ell_p$ sensitivity, $k$-subspace sensitivity is controlled by the *ridge leverage score*, defined as

$$\overline{\tau}_i(A) = a_i^\top (A^\top A + \lambda_{A_k} I)^\dagger a_i$$

where $\lambda_{A_k} = \|A - A_k\|_F^2/k$. The matrix in this definition is invertible unless $\mathrm{rank}(A) \le k$. For every $i \in [n]$, we have

$$s_i(A, A) \le (1 + 1/k) \cdot \overline{\tau}_i(A).$$

To see this, let $Q = I - P_x$ and $M = A^\top A + \lambda_{A_k} I$, and suppose $Qa_i \neq 0$. Applying the Cauchy–Schwarz inequality in the semi-inner product induced by $M$ with the unit vector $y = Qa_i/\|Qa_i\|_2$, we have $\|a_i^\top Q\|_2^2 \le \overline{\tau}_i(A) \cdot (\|Ay\|_2^2 + \lambda_{A_k}) \le (1 + 1/k)\, \overline{\tau}_i(A)\, \|AQ\|_F^2$. Here we use $\|Ay\|_2 \le \|AQ\|_F$ and $\lambda_{A_k} = \|A - A_k\|_F^2/k \le \|AQ\|_F^2/k$, where the latter holds because $AP_x$ has rank at most $k$. The ridge leverage score alone is not an overestimate of the sensitivity. For example, for $A = I_d$ and $k = 1$, we have $s_i(A, A) = 1/(d-1) > 1/d = \overline{\tau}_i(A)$. However, the value $2\overline{\tau}_i(A)$ is an overestimate, and the factor 2 is absorbed by the constant $c$ in the sampling probabilities of our framework. It is worth noting that the results of Cohen et al. [2017] used below are sampling guarantees for ridge leverage scores, and they do not pass through this bound. We then define the weights similarly to leverage scores:

$$w_i(A, B) = \begin{cases} a_i^\top (B^\top B + \lambda_{B_k} I)^\dagger a_i, & \text{if } a_i \in \mathrm{span}(B^\top B + \lambda_{B_k} I), \\ \infty, & \text{otherwise.} \end{cases}$$

We will explicitly define the notion of $\epsilon$-approximator:

**Definition C.20.** *Let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$, $\epsilon \in (0, 1)$ and $1 \le k \le d$. We say $B$ is an $\epsilon$-approximator of $A$ if*

- *$B$ is an $\epsilon$-coreset of $A$.*

- *The following additive-multiplicative spectral approximation guarantee holds:*

$$(1-\epsilon)B^\top B - \epsilon\lambda_{A_k} I \preceq A^\top A \preceq (1+\epsilon)B^\top B + \epsilon\lambda_{A_k} I.$$

The following two results due to Cohen et al. [2017] illustrate that an $\epsilon$-approximator indeed preserves all weights, and uniform sampling gives sufficiently good approximation.

**Lemma C.21** (Lemma 12 of Cohen et al. [2017][2])**.** *Let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$ and $\epsilon \in (0, 1/2]$. If $B$ is an $\epsilon$-approximator of $A$, then for any fixed $C$ and for all $i \in [n]$, $w_i(C, B) = (1 \pm 4\epsilon) \cdot w_i(C, A)$.*

**Lemma C.22** (Theorem 14 of Cohen et al. [2017])**.** *Let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$, $k \in [d]$ and $\epsilon, \delta \in (0, 1)$. Form $SA$ by including each point of $A$ independently with probability $\frac{1}{2}$. Let $q_i = \min\{1, c\epsilon^{-2} \log(k/\delta) \cdot w_i(A, SA)\}$ for a sufficiently large constant $c$, and let $B$ contain each $a_i$ independently with probability $q_i$, weighted by $1/q_i$. Then, with probability at least $1 - \delta$, $B$ is an $\epsilon$-approximator of $A$ of size $O(\epsilon^{-2} k \log(k/\delta))$.*

**Remark C.23.** *Cohen et al. [2017] state Theorem 14 for a constant accuracy, with the sampling probabilities $\min\{1, c \log(k/\delta) \cdot \min\{1, w_i(A, SA)\}\}$, and conclude that $B$ satisfies the hypotheses of their Lemma 12 for some constant $\epsilon$. However, their proof gives the $\epsilon$-dependence stated here. By their Theorem 11, which is monotonicity, we have $\min\{1, w_i(A, SA)\} \geq \overline{\tau}_i(A)$. Therefore, these overestimates are fed to their Lemmas 21 and 22, which are the independent-sampling forms of their Theorem 5 and Lemma 20 (the ingredient of their Theorem 6) stated for general $\epsilon$ with the probabilities $\min\{1, c\epsilon^{-2} \log(k/\delta)\widetilde{\tau}_i\}$. These probabilities coincide with our $q_i$ since $c\epsilon^{-2} \log(k/\delta) \geq 1$, so these lemmas yield the additive-multiplicative spectral approximation and the projection-cost preservation at accuracy $\epsilon$. We bound the number of sampled points through their Lemma 13 as in their proof. The reweighted columns contribute at most $3k/u = O(k \log(k/\delta))$ and the remaining ones at most $c\epsilon^{-2} \log(k/\delta) \cdot 12k$. This holds in expectation conditioned on the event that the reweighted half satisfies the hypotheses of their Lemma 12, an event of probability $1 - \delta/3$ in their proof, and by a Chernoff bound it also holds with probability $1 - \delta$.*

**Lemma C.24.** *Let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$, $k \leq d$, $\epsilon, \delta \in (0, 1)$, and define $w(A, B)$ as*

$$w_i(A, B) = \begin{cases} a_i^\top (B^\top B + \lambda_{B_k} I)^\dagger a_i, & \text{if } a_i \in \operatorname{span}(B^\top B + \lambda_{B_k} I), \\ \infty, & \text{otherwise}. \end{cases}$$

*Then $\sum_{i=1}^n w_i(A, A) \leq 2k$, and there exists a randomized algorithm (Algorithm 6) that implements* PREPROCESS *and* QUERY *procedures with*

- $\mathcal{T}_{\text{prep}}(s, d) = \widetilde{O}(ds^{\omega-1})$.
- $\mathcal{T}_{\text{query}}(s, d) = \widetilde{O}(d)$.

*Proof.* Let $A = U\Sigma V^\top$ be the SVD of $A$. Then,

$$\begin{aligned} \sum_{i=1}^n w_i(A, A) &= \sum_{i=1}^n a_i^\top (A^\top A + \lambda_{A_k} I)^\dagger a_i \\ &= \operatorname{tr}[A^\top A (A^\top A + \lambda_{A_k} I)^\dagger] \\ &= \operatorname{tr}[V\Sigma^2 V^\top (V\Sigma^2 V^\top + \lambda_{A_k} V V^\top)^\dagger] \\ &= \operatorname{tr}[\Sigma^2 (\Sigma^2 + \lambda_{A_k} I)^\dagger] \\ &= \sum_{i=1}^n \frac{\sigma_i^2}{\sigma_i^2 + \frac{\|A - A_k\|_F^2}{k}} \\ &\leq k + \sum_{i=k+1}^n \frac{\sigma_i^2}{\sigma_i^2 + \frac{\|A - A_k\|_F^2}{k}} \end{aligned}$$

[2]Note that while the original Lemma in Cohen et al. [2017] states the result in terms of ridge leverage score, their proof essentially shows that $B^\top B + \lambda_{B_k} I$ is a $1 \pm 4\epsilon$ spectral approximation of $A^\top A + \lambda_{A_k} I$, which gives the desired approximator guarantee.

**Algorithm 6** Classical oracle for ridge leverage score.

**data structure** RIDGELEVERAGESCORE
**members**
  $A \in \mathbb{R}^{n\times d}$
  $C \in \mathbb{R}^{s\times d}$
  $M \in \mathbb{R}^{O(\log n)\times d}$
  $N \in \mathbb{R}^{O(\log n)\times d}$
**end members**

**procedure** PREPROCESS($A \in \mathbb{R}^{n\times d}, C \in \mathbb{R}^{s\times d}$)
  $c \leftarrow 1000$
  Compute the compact SVD of $C$: $C = U\Sigma V^\top$ ▷ $r = \operatorname{rank}(C) \le s$, $\Sigma \in \mathbb{R}^{r\times r}$ invertible, $V \in \mathbb{R}^{d\times r}$
  $\lambda \leftarrow \frac{1}{k}\sum_{i=k+1}^{r}\sigma_i^2$ ▷ $\lambda = \lambda_{C_k} = \|C - C_k\|_F^2/k$, and $\lambda = 0$ iff $r \le k$
  Let $G \in \mathbb{R}^{c\log n\times r}$ and $G' \in \mathbb{R}^{c\log n\times d}$ be random Gaussian matrices
  $M \leftarrow G(\Sigma^2 + \lambda I)^{-1/2}V^\top$ ▷ $M \in \mathbb{R}^{c\log n\times d}$
  $N \leftarrow G'(I - VV^\top)$ ▷ $N \in \mathbb{R}^{c\log n\times d}$, sketch of the orthogonal complement of $\operatorname{span}(C^\top C)$
**end procedure**

**procedure** QUERY($i \in [n]$)
  **if** $\lambda = 0$ **and** $Na_i \neq 0$ **then return** $\infty$ ▷ $\operatorname{rank}(C) \le k$ and $a_i \notin \operatorname{span}(C^\top C)$, so $q_i = 1$
  **if** $\lambda > 0$ **then return** $\|Ma_i\|_2^2 + \lambda^{-1}\|Na_i\|_2^2$ ▷ sketch of $a_i^\top V(\Sigma^2 + \lambda I)^{-1}V^\top a_i + \lambda^{-1}\|(I - VV^\top)a_i\|_2^2$
  **return** $\|Ma_i\|_2^2$ ▷ $\lambda = 0$ and $a_i \in \operatorname{span}(C^\top C)$
**end procedure**
**end data structure**

$$
\begin{aligned}
&\le k + \sum_{i=k+1}^{n} \frac{\sigma_i^2}{\frac{\|A-A_k\|_F^2}{k}} \\
&= k + k \cdot \frac{\|A - A_k\|_F^2}{\|A - A_k\|_F^2} \\
&\le 2k
\end{aligned}
$$

where for the fifth step, we upper bound $\frac{\sigma_i^2}{\sigma_i^2 + \frac{\|A-A_k\|_F^2}{k}}$ by 1 for $i \le k$. The runtime analysis is identical to that of Lemma C.8, with the following additions for the correctness of Algorithm 6. Let $C = U\Sigma V^\top$ be the compact SVD of the current approximator $C$, where $r = \operatorname{rank}(C)$ and $V \in \mathbb{R}^{d\times r}$. Let $\lambda = \lambda_{C_k} = \frac{1}{k}\sum_{i>k}\sigma_i^2 = \|C - C_k\|_F^2/k$ be the value computed by PREPROCESS. If $\lambda > 0$, then $C^\top C + \lambda I$ is invertible and

$$(C^\top C + \lambda I)^{-1} = V(\Sigma^2 + \lambda I)^{-1}V^\top + \lambda^{-1}(I - VV^\top),$$

and hence $w_i(A, C) = \|(\Sigma^2 + \lambda I)^{-1/2}V^\top a_i\|_2^2 + \lambda^{-1}\|(I - VV^\top)a_i\|_2^2$. Both terms are sketched by Gaussian matrices with $O(\log n)$ rows. Thus, QUERY returns a $1 \pm \epsilon$ approximation of $w_i(A, C)$, which we turn into an overestimate by scaling. The second term vanishes when $r = d$, but it cannot be dropped whenever $r < d$. In our applications, the approximator has $s = \widetilde{O}(k)$ rows, typically far fewer than $d$. If $\lambda = 0$, i.e. $r \le k$, then the weight is $a_i^\top (C^\top C)^\dagger a_i$ for $a_i \in \operatorname{span}(C^\top C)$ and $\infty$ otherwise. In this case, QUERY handles the two subcases in the same way as Algorithm 3, using the same exact zero test on $Na_i$. Finally, computing $N = G' - (G'V)V^\top$ takes $\widetilde{O}(rd) \le \widetilde{O}(sd)$ time and each query takes $\widetilde{O}(d)$ time. Therefore, $\mathcal{T}_{\text{prep}}$ and $\mathcal{T}_{\text{query}}$ are as stated. □

One of the key features of the $\epsilon$-approximator for $k$-subspace approximation is that it is also an $\epsilon$-coreset.

**Theorem C.25.** *Let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$, $\epsilon, \delta \in (0, 1)$ and $k \in [d]$. There exists a quantum algorithm $\mathrm{QRLS}(A, k, \epsilon, \delta)$ that with probability at least $1 - \delta$, constructs an $\epsilon$-approximator of $A$*

*(in particular an $\epsilon$-coreset of $A$ for the $k$-subspace cost) with size $O(\epsilon^{-2}k\log(k/\delta))$, in expected time $\widetilde{O}(\epsilon^{-1}n^{0.5}dk^{0.5}+dk^{\omega-1})$ and, in expectation, $\widetilde{O}(\epsilon^{-1}n^{0.5}k^{0.5})$ queries to the points in $A$ (the expectations as in Theorem C.3).*

*Proof.* The proof is almost identical to the proof of Theorem C.12, except that the sizes are $s=\widetilde{O}(\epsilon^{-2}k)$ and $s'=\widetilde{O}(k)$. Let $T=O(\log n)$ be the number of levels. We run Algorithm 2 as QITERATESAMPLE$(A,\epsilon,\delta/2)$ with the oracle of Lemma C.24. Its $T+1$ calls of QSAMPLE are run with failure probability $\delta/(4T(T+1))$ each, which is at most $\delta/(4T)$ in total. This replaces the parameter $\delta'/(T+1)$ printed in Algorithm 2, although that parameter would serve as well. We replace items 2 and 3 of Assumption C.1 by Lemma C.22, which we apply at accuracy $0.01$ and failure probability $\delta/(4T)$ at each of the $T-1$ intermediate levels, and at accuracy $\epsilon$ and failure probability $\delta/4$ at the final level. By Lemma C.21, the oracle at level $t$ evaluates $w_i(A_t,C_{t-1})=\Theta(1)\cdot w_i(A_t,A_{t-1})$. The constant factors of Lemmas C.21 and C.24 are absorbed by the constant $c$ of Lemma C.22, and therefore its size bound applies at every level. As in Step 2 of the proof of Theorem C.12, the remaining $\delta/2$ covers the Gaussian sketches of the $T$ oracle preparations of Algorithm 6, each with $O(\log(n/\delta))$ rows and failure probability at most $\delta/(4T)$. It also covers the event $|A_0|\le 2\max\{s',3\log(4/\delta)\}$, which has probability at least $1-\delta/4$ by a Chernoff bound since $\mathbb{E}\,|A_0|\le s'$. The intersection of these events has probability at least $1-\delta$. On this intersection, the final level outputs an $\epsilon$-approximator of $A$ of size $O(\epsilon^{-2}k\log(k/\delta))$, and every intermediate sample, hence every set on which an oracle is prepared, has $O(s')$ points. Therefore, on this event each preparation costs $\mathcal{T}_{\mathrm{prep}}(s',d)=\widetilde{O}(dk^{\omega-1})$, since $s'=\widetilde{O}(k)$ and $k\le d$. Hence, when we plug these choices into Lemma C.24 and Theorem C.3, the hypothesis of the latter that $\mathcal{T}_{\mathrm{prep}}$ be linear is not needed, and we obtain a runtime of

$$\widetilde{O}(\epsilon^{-1}n^{0.5}dk^{0.5}+dk^{\omega-1}).$$

□

# D Quantum Column Subset Selection and Low-Rank Approximation

In this section, we present the first application of the generic sampling framework developed in Section C. In particular, when the cost is the $k$-subspace cost defined as $\mathrm{cost}(A,x)=\|(I-P_x)A\|_F^2$ where $x\in\mathcal{F}_k$, then an $\epsilon$-coreset of $A$ can be used to compute a Frobenius norm low-rank approximation. Let $A\in\mathbb{R}^{d\times n}$, we let the set of points be $\{a_1,\dots,a_n\}\subset\mathbb{R}^d$, and the goal is to compute a weighted subset of columns of $A$.

**Lemma D.1** (Lemma 3 of Cohen et al. [2015a])**.** *Let $A=\{a_1,\dots,a_n\}\subset\mathbb{R}^d$, $\epsilon\in(0,1)$, $k\in[\min\{n,d\}]$ and let $B\subset A$ be an $(\epsilon/3)$-coreset of $A$ with respect to the $k$-subspace cost. Then, the projection onto the top-$k$ left singular vectors of $B$, denoted by $P_{B_k}$, satisfies*

$$\|A-P_{B_k}P_{B_k}^\top A\|_F^2\le(1+\epsilon)\|A-A_k\|_F^2.$$

**Remark D.2.** *For a $\gamma$-coreset $B$, the two-sided guarantee gives $(1-\gamma)\|A-P_{B_k}P_{B_k}^\top A\|_F^2\le\|B-P_{B_k}P_{B_k}^\top B\|_F^2\le\|B-P_{A_k}P_{A_k}^\top B\|_F^2\le(1+\gamma)\|A-A_k\|_F^2$, which yields the factor $(1+\gamma)/(1-\gamma)$. This factor is at most $1+3\gamma$ for $\gamma\le 1/3$, and the statement above is this bound with $\gamma=\epsilon/3$. Moreover, the factor $(1+\gamma)/(1-\gamma)$ is attained up to any constant by two orthogonal columns of squared norms $a>b$ with coreset weights $1-\gamma$ and $1+\gamma$, once $(1+\gamma)b>(1-\gamma)a$.*

Cohen et al. [2017] showed that sampling according to ridge leverage scores gives a projection-cost preserving sample, and in fact a stronger $\epsilon$-approximator, which is an $\epsilon$-coreset. We discuss this in Section C.3, where we also show that the $k$-subspace sensitivity is at most $(1+1/k)$ times the ridge leverage score. We hence summarize the result below.

**Corollary D.3.** *Let $A\in\mathbb{R}^{d\times n}$, $\epsilon,\delta\in(0,1)$, $k\le\min\{n,d\}$ be a positive integer. There exists a quantum algorithm* QLOWRANKCMM$(A,k,\epsilon,\delta)$ *(the procedure* QRLS$(A,k,\epsilon,\delta)$ *of Theorem C.25 run on the columns of $A$) that constructs an $\epsilon$-coreset $C$ of $A$ for the $k$-subspace cost with probability at least $1-\delta$. The size of the coreset is at most $O(k\log(k/\delta)/\epsilon^2)$, the runtime is $\widetilde{O}(n^{0.5}dk^{0.5}\epsilon^{-1}+dk^{\omega-1})$, and the number of queries to the columns of $A$ is $\widetilde{O}(\epsilon^{-1}n^{0.5}k^{0.5})$.*

*Moreover, for any $s\ge k$ there is a fixed-size variant* QRLS$_s(A,k,s,\delta)$ *that, with probability at least $1-\delta$, returns constant-factor overestimates $\widetilde{\tau}_i$ of the rank-$k$ ridge leverage scores $\overline{\tau}_i(A)$ of the columns of $A$ with $\sum_i\widetilde{\tau}_i=O(k)$, together with a sample of the columns in which column $i$ is*

*included independently with probability* $q_i = \min\{1, s\cdot\widetilde{\tau}_i/k\}$ *(expected size* $O(s)$*), in* $\widetilde{O}(n^{0.5}d(k^{0.5}+s^{0.5})+dk^{\omega-1})$ *time and* $\widetilde{O}(n^{0.5}(k^{0.5}+s^{0.5}))$ *queries to the columns of* $A$*.*

*Proof.* The coreset statement follows from applying Theorem C.25 to the columns of $A$. The fixed-size variant is obtained as for QLS in Theorem C.12, with the oracle of Lemma C.24. The recursion of Theorem C.3 at constant accuracy and failure probability $\delta/2$ yields an $O(1)$-approximator $C_T$ of size $s' = \widetilde{O}(k)$ in $\widetilde{O}(n^{0.5}dk^{0.5}+dk^{\omega-1})$ time and $\widetilde{O}(n^{0.5}k^{0.5})$ column queries. By Lemma C.21, we have $w_i(A, C_T) = \Theta(1)\cdot\overline{\tau}_i(A)$ for every $i$, and Algorithm 6 evaluates this weight up to a constant factor in $\widetilde{O}(d)$ time per query. After scaling by a constant, we obtain overestimates $\widetilde{\tau}_i$ with $\sum_i\widetilde{\tau}_i = O(\sum_i\overline{\tau}_i(A)) = O(k)$ by Lemma C.24. Finally, Lemma B.15 with failure probability $\delta/2$ draws the sample, with $\sum_i q_i = O(s)$, in $\widetilde{O}(\sqrt{ns}\cdot d)$ time and $\widetilde{O}(\sqrt{ns})$ column queries. □

We note that in addition, $C$ is a column subset selection of $A$:

**Definition D.4** (Rank-$k$ Column Subset Selection)**.** *For* $n' \le n$*, a subset of* $A$*'s columns* $C \in \mathbb{R}^{d\times n'}$ *(possibly with repetitions) is a* $(1+\epsilon)$ *factor column subset selection if there exists a rank-*$k$ *matrix* $X \in \mathbb{R}^{n'\times n}$ *with*

$$\|A - CX\|_F^2 \le (1+\epsilon)\|A - A_k\|_F^2.$$

Note that for column subset selection alone, sampling by ridge leverage scores only needs an oversampling factor of $1/\epsilon$, rather than $1/\epsilon^2$:

**Lemma D.5** (Theorem 7 of Cohen et al. [2017])**.** *Let* $A \in \mathbb{R}^{d\times n}$*,* $k \le \min\{n, d\}$*,* $\epsilon, \delta \in (0,1)$*, and let* $\widetilde{\tau}_i \ge \overline{\tau}_i(A)$ *be overestimates of the rank-*$k$ *ridge leverage scores of the columns of* $A$*. Let* $C$ *consist of* $t = O(\epsilon^{-1}(\log k + \log(1/\delta))\sum_i\widetilde{\tau}_i)$ *columns of* $A$*, drawn independently with replacement, each with probability proportional to* $\widetilde{\tau}_i$*. Then, with probability at least* $1-\delta$*,* $\|A - (CC^\dagger A)_k\|_F^2 \le (1+\epsilon)\|A - A_k\|_F^2$*, and since* $(CC^\dagger A)_k$ *has rank* $k$ *and lies in the column span of* $C$*,* $C$ *is a* $(1+\epsilon)$ *factor rank-*$k$ *column subset selection of* $A$ *(Definition D.4).*

When the $\widetilde{\tau}_i$ are constant-factor approximations of $\overline{\tau}_i(A)$, by Lemma C.24 we have $\sum_i\widetilde{\tau}_i = O(k)$, and hence $t = O(\epsilon^{-1}k\log(k/\delta))$. Cohen et al. [2017] state Theorem 7 for $t$ draws with replacement, and their Appendix B writes out independent sampling only for the $\epsilon^{-2}$ forms of their Theorem 5 and Lemma 20. The following lemma transfers Theorem 7 to the independent sampling of $\mathrm{QRLS}_s$.

**Lemma D.6** (Bernoulli form of Theorem 7 of Cohen et al. [2017])**.** *Under the hypotheses of Lemma D.5, let* $t$ *be as there, with the constant in* $t$ *large enough that* $t \ge 4\log(1/\delta)$*, and let* $p_i = \widetilde{\tau}_i/\sum_j\widetilde{\tau}_j$*. Let* $B$ *be a random subset of the columns of* $A$ *in which column* $i$ *is included independently with probability* $q_i \ge \min\{1, 2tp_i\}$*. Then* $\mathbb{E}|B| = \sum_i q_i$*, and with probability at least* $1-2\delta$*,* $B$ *is a* $(1+\epsilon)$ *factor rank-*$k$ *column subset selection of* $A$ *(Definition D.4). In particular, the inclusion probabilities* $\min\{1, s\,\widetilde{\tau}_i/k\}$ *of* $\mathrm{QRLS}_s(A, k, s, \delta')$ *satisfy the hypothesis whenever* $s \ge 2tk/\sum_j\widetilde{\tau}_j = O(\epsilon^{-1}k(\log k + \log(1/\delta)))$*, a threshold that does not depend on* $\sum_j\widetilde{\tau}_j$*.*

*Proof.* We first note that adding columns preserves column subset selection. Indeed, if every column of $C$ occurs in $C'$ and $\mathrm{rank}(X) \le k$, then summing the rows of $X$ belonging to the draws of each column of $C'$ gives $X'$ with $C'X' = CX$ and $\mathrm{rank}(X') \le k$. Let $X_1, X_2, \ldots$ be independent draws from the distribution $(p_i)_{i\in[n]}$, let $N \sim \mathrm{Poisson}(2t)$ be independent of them, and let $S_N = \{X_1, \ldots, X_N\} \subseteq [n]$. By Poisson splitting, the events $\{i \in S_N\}$ are independent with $\Pr[i \in S_N] = 1 - e^{-2tp_i} \le \min\{1, 2tp_i\} \le q_i$. Therefore, adding each $i \notin S_N$ independently with probability $r_i := \big(q_i - (1-e^{-2tp_i})\big)/e^{-2tp_i} \in [0,1]$ gives a set $B \supseteq S_N$ with the law of the statement, and $\mathbb{E}|B| = \sum_i q_i$. By Lemma D.5, with probability at least $1-\delta$, the columns $a_{X_1}, \ldots, a_{X_t}$ are a $(1+\epsilon)$ factor rank-$k$ column subset selection of $A$. If this event and the event $\{N \ge t\}$ both hold, then $S_N \supseteq \{X_1, \ldots, X_t\}$, and hence, by monotonicity, $B \supseteq S_N$ is also such a selection. Applying Markov's inequality to $e^{-\theta N}$ with $\theta = \log 2$, we have $\Pr[N < t] \le \Pr[N \le t] \le e^{-t}2^t = (2/e)^t \le \delta$, since $t \ge 4\log(1/\delta) \ge \log(1/\delta)/\log(e/2)$. A union bound gives the probability $1-2\delta$. For $A \ne 0$, let $\sigma_1 \ge \cdots \ge \sigma_r > 0$ be the nonzero singular values of $A$ and let $\lambda = \|A - A_k\|_F^2/k$. Then, as in the proof of Lemma C.24, we have $\sum_i\widetilde{\tau}_i \ge \sum_i\overline{\tau}_i(A) = \sum_{j=1}^r\sigma_j^2/(\sigma_j^2+\lambda)$, and this sum is at least $1/2$. Indeed, if $r \le k$, then $\lambda = 0$ and every term equals 1. If $r > k$ and $\sigma_k^2 \ge \lambda$, then the $k$-th term is at least $1/2$. If $r > k$ and $\sigma_k^2 < \lambda$, then $\sigma_j^2 + \lambda < 2\lambda$ for $j > k$, and thus these terms sum

to at least $\sum_{j>k}\sigma_j^2/(2\lambda) = k/2$. Consequently, $t \geq c\,\epsilon^{-1}\log(1/\delta)\sum_i \widetilde{\tau}_i \geq 4\log(1/\delta)$ once the constant $c$ in $t$ is at least 8. For $A = 0$, every column subset is a $(1+\epsilon)$ factor selection. For the last claim, $\min\{1, s\,\widetilde{\tau}_i/k\} \geq \min\{1, 2tp_i\}$ for all $i$ as soon as $s/k \geq 2t/\sum_j \widetilde{\tau}_j$, and by the definition of $t$, we have $2tk/\sum_j \widetilde{\tau}_j = O(\epsilon^{-1}(\log k + \log(1/\delta))) \cdot k$.

□

The weights $1/q_i$ that $\mathrm{QRLS}_s$ attaches to the sampled columns play no role in the column subset selection property, since the matrix $X$ in Definition D.4 is unconstrained apart from its rank. We utilize this fact to further derive an algorithm for outputting a low-rank approximation of $A$, which could subsequently be generalized to tensors. We state a tool for solving a bilinear multiple response regression.

**Lemma D.7** (Generalized Low-Rank Approximation [Friedland and Torokhti, 2007])**.** *Let $A \in \mathbb{R}^{d\times n}$, $B \in \mathbb{R}^{d\times p}$ and $C \in \mathbb{R}^{q\times n}$, let $k \leq \min\{n, d\}$ be a positive integer. The following bilinear regression problem*

$$\min_{X\in\mathbb{R}^{p\times q}:\mathrm{rank}(X)\leq k} \|A - BXC\|_F^2$$

*is minimized by $X_* = B^\dagger[P_B A P_C]_k C^\dagger$ where $P_B, P_C$ are the projection matrices onto $B, C$ respectively.*

**Algorithm 7** Quantum low-rank approximation.

**procedure** QLowRank($A \in \mathbb{R}^{d\times n}, k, \epsilon$)
  $k_1 \leftarrow O(\epsilon^{-1}k\log k)$ ▷ Lemma D.6 at $\delta = 0.0005$
  $k_2 \leftarrow O(k_1\log k_1 + \epsilon^{-1}k_1)$
  $k_2' \leftarrow O(\epsilon^{-2}k_1\log k_1)$
  $k_3 \leftarrow O(\epsilon^{-2}k_2\log k_2)$
  $C \leftarrow \mathrm{QRLS}_s(A, k, k_1, 0.001)$ ▷ $C \in \mathbb{R}^{d\times k_1}$, a $(1+\epsilon)$ rank-$k$ column subset selection (Lemma D.6)
  $S \leftarrow \mathrm{QLS}(C, k_2, 0.001)$ ▷ $S \in \mathbb{R}^{k_2\times d}$, Theorem C.12.
  $T_1 \leftarrow \mathrm{QLS}(C, k_2', 0.001)$ ▷ $T_1 \in \mathbb{R}^{k_2'\times d}$, Theorem C.12.
  $T_2 \leftarrow \mathrm{QLS}(SA, k_3, 0.001)$ ▷ $T_2 \in \mathbb{R}^{n\times k_3}$, Theorem C.12.
  Solve $\min_{Y\in\mathbb{R}^{k_1\times k_2}:\mathrm{rank}(Y)\leq k} \|T_1CYSAT_2 - T_1AT_2\|_F^2$ by the lemma of Friedland and Torokhti [2007]:
  $\widehat{M} \leftarrow (T_1C)^\dagger[P_{T_1C}T_1AT_2P_{SAT_2}]_k(SAT_2)^\dagger$ ▷ $\widehat{M} \in \mathbb{R}^{k_1\times k_2}$ and $\mathrm{rank}(\widehat{M}) \leq k$.
  Write $\widehat{M}$ into factored form $\widehat{M} = \widehat{X}\widehat{Y}$ ▷ $\widehat{X} \in \mathbb{R}^{k_1\times k}, \widehat{Y} \in \mathbb{R}^{k\times k_2}$.
  **return** $C\widehat{X}$, $\widehat{Y}SA$ in factored form
**end procedure**

**Theorem D.8.** *Let $A \in \mathbb{R}^{d\times n}$ and $\epsilon \in (0, 0.1)$, and let $k \leq \min\{d, n\}$ be a positive integer. Then, there exists a quantum algorithm (Algorithm 7) that outputs a pair of matrices $M \in \mathbb{R}^{d\times k}, N \in \mathbb{R}^{n\times k}$ of rank at most $k$, represented in factored form (Remark D.9), such that*

$$\|A - MN^\top\|_F^2 \leq (1+\epsilon)\cdot\|A - A_k\|_F^2$$

*holds with probability at least 0.99. Moreover, Algorithm 7 runs in time*

$$\widetilde{O}(\epsilon^{-0.5}n^{0.5}dk^{0.5} + dk^{\omega-1} + \epsilon^{-2.5}d^{0.5}k^{1.5} + \epsilon^{-4}n^{0.5}k^{1.5} + \mathrm{poly}(k/\epsilon)),$$

*and uses $\widetilde{O}(\epsilon^{-2}n^{0.5}k^{0.5})$ queries to the columns and $\widetilde{O}(\epsilon^{-1.5}d^{0.5}k^{0.5})$ queries to the rows of $A$.*

**Remark D.9.** *The factors are returned implicitly. The factor $M = C\widehat{X}$ is specified by the $k_1$ sampled columns of $A$ together with $\widehat{X}$, and $N^\top = \widehat{Y}SA$ is specified by the row indices and weights of $S$ together with $\widehat{Y}$. Any entry of $M$ or $N$ can be computed on demand in $\widetilde{O}(\mathrm{poly}(k/\epsilon))$ time, and this is the same output convention as our kernel low-rank approximation result. Materializing the support $SA$ of the second factor costs an additional $\widetilde{O}(nk_2) = \widetilde{O}(\epsilon^{-2}nk)$ time, and forming $M$ and $N$ as explicit dense matrices by the naive products $C\widehat{X}$ and $(\widehat{Y}SA)^\top$ costs $\widetilde{O}(\epsilon^{-1}dk^2 + \epsilon^{-2}nk^2)$. The cost of the first product would dominate the leading term unless $n \gtrsim \epsilon^{-1}k^3$, which our assumption on the dimensions gives, and that of the second product unless $d \gtrsim \epsilon^{-1.5}n^{0.5}k^{1.5}$. In contrast, no cost linear in $n$ is incurred in factored form.*

*Proof.* Write $\mathrm{OPT} := \|A - A_k\|_F^2$. We first prove correctness, in three steps.

*Step 1: a good column subset.* Suppose that $\mathrm{QRLS}_s(A, k, k_1, 0.001)$ succeeds. By Corollary D.3, the scores $\widetilde{\tau}_i$ it computes are constant-factor overestimates of $\overline{\tau}_i(A)$ with $\sum_i \widetilde{\tau}_i = O(k)$, and $C$ includes column $i$ independently with probability $\min\{1, k_1\widetilde{\tau}_i/k\}$. We apply Lemma D.6 with $\delta = 0.0005$, and by its last claim, the hypothesis $k_1 \geq 2tk/\sum_j \widetilde{\tau}_j$ holds when the constant in $k_1 = O(\epsilon^{-1}k\log k)$ is large enough. Consequently, with probability at least $1 - 2\delta = 0.999$, $C$ is a $(1+\epsilon)$ factor rank-$k$ column subset selection in the sense of Definition D.4, meaning that there is a rank-$k$ matrix $X^\star \in \mathbb{R}^{k_1 \times n}$ with $\|A - CX^\star\|_F^2 \leq (1+\epsilon)\,\mathrm{OPT}$.

*Step 2: a good rank-$k$ solution in the row span of $SA$ exists.* Let $Z := CX^\star \in \mathbb{R}^{d\times n}$, which has rank at most $k$ and whose column span is contained in that of $C$. Therefore, for every row index $i \in [d]$, the leverage score of the $i$-th row of $Z$ is at most that of the $i$-th row of $C$, since both are the $i$-th diagonal entries of the orthogonal projectors onto the respective column spans and the projector for $Z$ is dominated by the one for $C$ in the positive semidefinite order. The matrix $S$ samples rows independently with probabilities $\min\{1, c(\log k_1 + \epsilon^{-1})\cdot\widetilde{\tau}_i\}$, where $\widetilde{\tau}_i = \Theta(\tau_i(C))$ are computed by QLS in Theorem C.12. For $c$ large enough, these probabilities dominate $\min\{1, c'(\log k + \epsilon^{-1})\tau_i(Z)\}$, and since $\sum_i \widetilde{\tau}_i = O(k_1)$, the expected number of sampled rows is $k_2 = O(k_1\log k_1 + \epsilon^{-1}k_1)$. The proof of Lemma B.13 only uses that the sampling probabilities dominate the leverage scores of the design, with the sample size governed by their sum. Thus, we can apply this lemma to the design $Z$ with these probabilities in place of the leverage scores of $Z$, and for $Y_S := (SZ)^\dagger SA$ this gives

$$\|ZY_S - A\|_F^2 \leq (1+\epsilon)\cdot\min_Y \|ZY - A\|_F^2 \leq (1+\epsilon)\cdot\|Z - A\|_F^2 \leq (1+\epsilon)^2\,\mathrm{OPT},$$

where the second inequality takes $Y = I$. Since $ZY_S = C\cdot X^\star(SCX^\star)^\dagger\cdot SA = C\widehat{Y}_0 SA$ with $\widehat{Y}_0 := X^\star(SCX^\star)^\dagger \in \mathbb{R}^{k_1\times k_2}$ of rank at most $k$, we have

$$\min_{Y\in\mathbb{R}^{k_1\times k_2}:\mathrm{rank}(Y)\leq k} \|A - CYSA\|_F^2 \leq (1+\epsilon)^2\,\mathrm{OPT}\,.$$

*Step 3: finding such a solution.* Write $f(Y) := \|CYSA - A\|_F^2$, $g(Y) := \|T_1(CYSA - A)\|_F^2$, and $h(Y) := \|T_1(CYSA - A)T_2\|_F^2$. The matrix $T_1$ samples $k_2' = O(\epsilon^{-2}k_1\log k_1)$ rows of $C$ according to leverage scores, $C$ has rank at most $k_1$, and $CYSA$ is always of the form $CX$. Therefore, by Lemma B.12 with design $C$ and response $A$, there is a constant $c_1$ such that $g(Y) = (1\pm\epsilon)f(Y) + c_1$ for all $Y$. Similarly, $T_2$ samples $k_3 = O(\epsilon^{-2}k_2\log k_2)$ columns of $SA$ according to the leverage scores of the rows of $(SA)^\top$, and $SA$ has rank at most $k_2$. Applying Lemma B.12 with design $(SA)^\top$ and response $(T_1A)^\top$, we obtain a constant $c_2$ such that $\|(WSA - T_1A)T_2\|_F^2 = (1\pm\epsilon)\|WSA - T_1A\|_F^2 + c_2$ for all $W$, and taking $W = T_1CY$ gives $h(Y) = (1\pm\epsilon)g(Y) + c_2$ for all $Y$. Let $\widehat{M}$ minimize $h$ over $\{Y : \mathrm{rank}(Y) \leq k\}$. By the lemma of Friedland and Torokhti [2007], we have $\widehat{M} = (T_1C)^\dagger[P_{T_1C}T_1AT_2P_{SAT_2}]_k(SAT_2)^\dagger$, which the algorithm computes, and $\mathrm{rank}(\widehat{M}) \leq k$. Comparing with $\widehat{Y}_0$ of Step 2, we have $h(\widehat{M}) \leq h(\widehat{Y}_0)$, and since the constant $c_2$ cancels, we obtain $(1-\epsilon)g(\widehat{M}) \leq (1+\epsilon)g(\widehat{Y}_0)$. Let $\widehat{B} = CC^\dagger A - A$ be the residual of the unconstrained problem $\min_X \|CX - A\|_F^2$. The constant $c_1 = \|T_1\widehat{B}\|_F^2 - \|\widehat{B}\|_F^2$ does not cancel, but it is small. Since $T_1$ samples with rescaling, $\|T_1\widehat{B}\|_F^2$ is an unbiased estimator of $\|\widehat{B}\|_F^2$, and hence, by Markov's inequality, $\|T_1\widehat{B}\|_F^2 \leq 1000\|\widehat{B}\|_F^2$ with probability at least $0.999$. Consequently, $|c_1| \leq 999\|\widehat{B}\|_F^2 \leq 999 f(\widehat{Y}_0)$, where the last inequality holds because the unconstrained minimum is at most $f(\widehat{Y}_0)$. Substituting $g = (1\pm\epsilon)f + c_1$ on both sides, we have

$$(1-\epsilon)^2 f(\widehat{M}) \leq (1+\epsilon)^2 f(\widehat{Y}_0) + 2\epsilon\cdot|c_1| \leq \big((1+\epsilon)^2 + 1998\epsilon\big)\cdot f(\widehat{Y}_0),$$

which implies $f(\widehat{M}) \leq (1+O(\epsilon))\cdot(1+\epsilon)^2\,\mathrm{OPT}$, and rescaling $\epsilon$ by a constant gives $\|A - C\widehat{M}SA\|_F^2 \leq (1+\epsilon)\,\mathrm{OPT}$. Moreover, the output $M = C\widehat{X}$, $N^\top = \widehat{Y}SA$ with $\widehat{X}\widehat{Y} = \widehat{M}$ has rank at most $k$. The call to $\mathrm{QRLS}_s$, the three QLS calls, Lemma D.6 applied with $\delta = 0.0005$, Lemma B.13, the two applications of Lemma B.12, and the Markov step each fail with probability at most $0.001$, and hence, by a union bound, the success probability is at least $0.99$.

For the running time, by the fixed-size variant of Corollary D.3 with $s = k_1$, generating $C$ takes $\widetilde{O}(\epsilon^{-0.5}n^{0.5}dk^{0.5} + dk^{\omega-1})$ time. Generating $S$ takes $\widetilde{O}(\sqrt{dk_2}k_1 + k_1^\omega) = \widetilde{O}(\epsilon^{-2}d^{0.5}k^{1.5} + \epsilon^{-\omega}k^\omega)$

time, and generating $T_1$ takes $\widetilde{O}(\sqrt{dk_2'}k_1 + k_1^\omega) = \widetilde{O}(\epsilon^{-2.5}d^{0.5}k^{1.5} + \epsilon^{-\omega}k^\omega)$ time, using $k_1 = \widetilde{O}(\epsilon^{-1}k)$, $k_2 = \widetilde{O}(\epsilon^{-2}k)$, and $k_2' = \widetilde{O}(\epsilon^{-3}k)$. We never materialize $SA$, since the algorithm only accesses $SAT_2$, a $\mathrm{poly}(k/\epsilon)$-size submatrix of $A$, and the factor $\widehat{Y}SA$ is returned implicitly as in Remark D.9. Generating $T_2$ takes $\widetilde{O}(\sqrt{nk_3}k_2 + k_2^\omega) = \widetilde{O}(\epsilon^{-4}n^{0.5}k^{1.5} + \epsilon^{-2\omega}k^\omega)$ time, using $k_3 = \widetilde{O}(\epsilon^{-4}k)$. Finally, computing $T_1C$, $SAT_2$, their pseudoinverses, and their projections takes $\mathrm{poly}(k/\epsilon)$ time, since forming these matrices is simply selecting and rescaling entries, and the resulting matrices are of size $\mathrm{poly}(k/\epsilon)$. The matrix $T_1AT_2$ is obtained in the same way, hence $\widehat{M}$ can be computed in $\mathrm{poly}(k/\epsilon)$ time. Summing up these bounds gives the running time stated in the theorem. For the query counts: generating $C$ uses $\widetilde{O}(n^{0.5}(k^{0.5} + k_1^{0.5})) = \widetilde{O}(\epsilon^{-0.5}n^{0.5}k^{0.5})$ queries to columns of $A$, the Grover searches for $S$ and $T_1$ use $\widetilde{O}(\sqrt{dk_2} + \sqrt{dk_2'}) = \widetilde{O}(\epsilon^{-1.5}d^{0.5}k^{0.5})$ queries to rows, and the search for $T_2$ uses $\widetilde{O}(\sqrt{nk_3}) = \widetilde{O}(\epsilon^{-2}n^{0.5}k^{0.5})$ queries to columns. □

# E Quantum Kernel Low-Rank Approximation

Given a set of points $\{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ and a positive definite kernel function $\mathsf{K} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, the kernel low-rank approximation problem asks to find a pair $M, N \in \mathbb{R}^{n\times k}$ such that $\|K - MN^\top\|_F^2 \le (1+\epsilon)\cdot\|K - K_k\|_F^2$, where $K \in \mathbb{R}^{n\times n}$ is the kernel matrix induced by $\mathsf{K}$, with $K_{i,j} = \mathsf{K}(x_i, x_j)$. Note that explicitly forming the matrix $K$ takes $\Omega(n^2)$ evaluations of $\mathsf{K}(\cdot,\cdot)$, which is usually too expensive to be afforded. Since $\mathsf{K}$ is positive definite, there exists a feature mapping $\phi : \mathbb{R}^d \to \mathbb{R}^{d'}$ such that $K = \Phi\Phi^\top$ where $\Phi \in \mathbb{R}^{n\times d'}$ with the $i$-th row being $\phi(x_i)$. Musco and Musco [2017] gives a low-rank approximation for $\Phi$ using $\widetilde{O}(\epsilon^{-1}nk)$ evaluations of $\mathsf{K}(\cdot,\cdot)$ and an additional $\widetilde{O}(\epsilon^{-2}nk^2)$ time (their Table 1 and Theorem 17). Musco and Woodruff [2017] show that the above guarantee for $K$ can be achieved with $\widetilde{O}(\epsilon^{-2.5}nk)$ kernel evaluations and an additional $\widetilde{O}(\epsilon^{-2(\omega-1)}nk^{\omega-1})$ time, and Bakshi et al. [2020] with $\widetilde{O}(\epsilon^{-1}nk)$ kernel evaluations and an additional $\widetilde{O}(\epsilon^{-(\omega-1)}nk^{\omega-1})$ time[3]. In this section, we will present a quantum algorithm based on the techniques developed in Sections C and D, that computes a low-rank approximation for the kernel matrix using *a sublinear number of kernel evaluations and additional operations.*

Before diving into our main result, we introduce some notations. We will extensively use $KD_1$ or $D_2^\top KD_1$ to denote a weighted sampling of $K$, in particular,

- If $D \in \mathbb{R}^{n\times t}$, we use $D^\top K_i \in \mathbb{R}^t$ to denote the vector $v$ with $v_j := D(j)\cdot\mathsf{K}(x_i, x_j)$, where $j \in D$ is the $j$-th sample of $D$, and $D(j)$ is the corresponding weight.
- If $D \in \mathbb{R}^{n\times t}$, we use "$KD$ in factored form" to denote a data structure that when the $i$-th row is queried, computes $v \in \mathbb{R}^t$ where $v_j := D(j)\cdot\mathsf{K}(x_i, x_j)$ for $j \in D$.
- If $D_1 \in \mathbb{R}^{n\times t_1}$ and $D_2 \in \mathbb{R}^{n\times t_2}$, we use "$D_2^\top KD_1$ in factored form" to denote a data structure that supports queries to either row or column, where for $i$-th row, it computes a vector $v^{\mathrm{row}} \in \mathbb{R}^{t_1}$ where $v_j^{\mathrm{row}} := D_1(j)D_2(i)\cdot\mathsf{K}(x_i, x_j)$ for $j \in D_1$ and $i \in D_2$. Similarly the operation applies to the column.
- Sometimes given $KD \in \mathbb{R}^{n\times t_1}$ in factored form, we will compose it with another matrix $M \in \mathbb{R}^{t_1\times t_2}$. We then use "$KDM$ in factored form" to denote a data structure that supports row queries, such that when the $i$-th row is queried, it returns $M^\top v$ where $v_j := D(j)\cdot\mathsf{K}(x_i, x_j)$ for $j \in D$.

**Lemma E.1** (Guarantee of Algorithm 9)**.** *Let $K$ be the kernel matrix of $X = \{x_1, \ldots, x_n\}$ and let $\Phi := K^{1/2}$, a symmetric matrix with $\Phi^\top\Phi = K$ whose rows $\phi_i$ satisfy $\phi_i^\top\phi_j = \mathsf{K}(x_i, x_j)$. Let $s \ge 1$, $\delta \in (0,1)$, and let $k$ be the integer set by* QNyströmKernel$(X, \mathsf{K}, s, \delta)$*. With probability at least $1-\delta$, the procedure returns a sample $D$ of $[n]$ in which $i$ is included independently with probability $p_i = \min\{1, 16\log(2k/\delta)\,q_i\}$ and weight $p_i^{-1/2}$, where the values $q_i$ of its last oracle satisfy*

$$\overline{\tau}_i(\Phi) \le q_i \le 6\,\overline{\tau}_i(\Phi) \qquad \text{for every } i \in [n],$$

[3]Note that Musco and Woodruff [2017], Bakshi et al. [2020] phrase their algorithm as a low-rank approximation for PSD matrix $A$, and their runtime is stated in terms of reads to $A$. Observe that a read to an entry of $A$ could be translated to one kernel evaluation.

**Algorithm 8** Quantum kernel low-rank approximation.

**procedure** QLOWRANKKERNEL($\{x_1, \dots, x_n\} \in (\mathbb{R}^d)^n, \mathsf{K} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}, k, \epsilon, \delta$)
  $c \leftarrow 1000$
  $t \leftarrow c\sqrt{\frac{nk}{\epsilon}} \log(n/\delta)$
  $k' \leftarrow 100(k/\epsilon)\log(12k/(\epsilon\delta))$ ▷ $k' = \widetilde{O}(k/\epsilon)$
  $D_1 \leftarrow$ QNYSTRÖMKERNEL($\{x_1, \dots, x_n\}, \mathsf{K}, k', \delta/6$) with the probabilities of its last sampling step replaced by $p_i = \min\{1, \sqrt{\epsilon n/k} \cdot 16 q_i \log(12n/\delta)\}$ ▷ $D_1 \in \mathbb{R}^{n \times t}$
  $D_2 \leftarrow$ QNYSTRÖMKERNEL($\{x_1, \dots, x_n\}, \mathsf{K}, k', \delta/6$) with the probabilities of its last sampling step replaced by $p_i = \min\{1, \sqrt{\epsilon n/k} \cdot 16 q_i \log(12n/\delta)\}$ ▷ $D_2 \in \mathbb{R}^{n \times t}$
  $C \leftarrow KD_1$ in factored form ▷ $C \in \mathbb{R}^{n \times t}$.
  $R \leftarrow D_2^\top K D_1$ in factored form ▷ $R \in \mathbb{R}^{t \times t}$.
  $\epsilon_0 \leftarrow 0.01$
  $\widetilde{R} \leftarrow$ QLOWRANKCMM($R^\top, k/\epsilon, \epsilon_0, \delta/6$) ▷ Corollary D.3 on the columns of $R^\top$
  $Z \leftarrow$ top-$k/\epsilon$ left singular vectors of $\widetilde{R}$ ▷ $Z \in \mathbb{R}^{t \times k/\epsilon}$, orthonormal columns
  ▷ Solve the regression $\min_{W \in \mathbb{R}^{n \times k/\epsilon}} \|C - WZ^\top\|$.
  Implement oracle for $p_i = \min\{1, \frac{\epsilon k'}{k} \cdot \|z_i\|_2^2\}$, $i \in [t]$ ▷ $\sum_{i=1}^t p_i \le k'$
  $D_3 \leftarrow$ QSAMPLE($p, 0.001$) ▷ $D_3 \in \mathbb{R}^{t \times k'}$ samples rows of $Z$
  ▷ Solve the surrogate regression $\min_W \|CD_3 - WZ^\top D_3\|$.
  $W \leftarrow CD_3(Z^\top D_3)^\dagger$ in factored form ▷ $W = K\left(D_1 D_3 (Z^\top D_3)^\dagger\right) \in \mathbb{R}^{n \times k/\epsilon}$
  ▷ Solve the regression $\min_{Y : \mathrm{rank}(Y) \le k} \|K - WYW^\top\|_F^2$.
  $D_4 \leftarrow$ QLS($W, k'/\epsilon^2, \delta/6$) ▷ $D_4 \in \mathbb{R}^{n \times k'/\epsilon^2}$, sample rows.
  $D_5 \leftarrow$ QLS($W, k'/\epsilon^2, \delta/6$) ▷ $D_5 \in \mathbb{R}^{n \times k'/\epsilon^2}$, sample columns.
  Compute $D_4^\top W$ and $W^\top D_5$ ▷ $D_4^\top W \in \mathbb{R}^{k'/\epsilon^2 \times k/\epsilon}$, $W^\top D_5 \in \mathbb{R}^{k/\epsilon \times k'/\epsilon^2}$.
  $P_{D_4^\top W} \leftarrow D_4^\top W (W^\top D_4 D_4^\top W)^\dagger W^\top D_4$, $P_{W^\top D_5} \leftarrow D_5^\top W (W^\top D_5 D_5^\top W)^\dagger W^\top D_5$ ▷ projectors onto $\mathrm{col}(D_4^\top W)$ and $\mathrm{row}(W^\top D_5)$
  Compute $D_4^\top K D_5$ ▷ $D_4^\top K D_5 \in \mathbb{R}^{k'/\epsilon^2 \times k'/\epsilon^2}$.
  Compute $[P_{D_4^\top W}(D_4^\top K D_5) P_{W^\top D_5}]_k$ ▷ Of rank $k$.
  $Y_* \leftarrow (D_4^\top W)^\dagger [P_{D_4^\top W}(D_4^\top K D_5) P_{W^\top D_5}]_k (W^\top D_5)^\dagger$ ▷ $Y_* \in \mathbb{R}^{k/\epsilon \times k/\epsilon}$ of rank $k$.
  $U_* \leftarrow$ top-$k$ left singular vectors of $Y_*$ ▷ $U_* \in \mathbb{R}^{k/\epsilon \times k}$.
  $D_6 \leftarrow$ QLS($WU_*, c(k \log k + k/\epsilon), \delta/6$) ▷ $D_6 \in \mathbb{R}^{n \times O(k \log k + k/\epsilon)}$.
  ▷ Solve the regression $\min_{N \in \mathbb{R}^{n \times k}} \|D_6^\top K - D_6^\top WU_* N^\top\|_F^2$.
  $N^\top \leftarrow (D_6^\top WU_*)^\dagger (D_6^\top K)$ ▷ $N \in \mathbb{R}^{n \times k}$ in factored form
  **return** $M := WU_*$ and $N$ in factored form
**end procedure**

*$\overline{\tau}_i(\Phi) = \phi_i^\top (K + \lambda_{\Phi_k} I)^{-1} \phi_i$ being the rank-$k$ ridge leverage scores of the rows of $\Phi$ (Definition B.7), and in particular $\sum_i q_i \le 12k$, and the sample has expected size $O(k \log(k/\delta)) = O(s)$. If the probabilities of the last sampling step are replaced by $\min\{1, \beta \cdot 16 \log(2n/\delta)\, q_i\}$ for some $\beta \ge 1$, the same holds with expected size $O(\beta k \log(n/\delta))$. The procedure runs in expected time $\widetilde{O}(\sqrt{ns}\,(s\mathcal{T}_\mathsf{K} + s^2) + s^2\mathcal{T}_\mathsf{K} + s^\omega)$ and makes $\widetilde{O}(\sqrt{ns} \cdot s)$ kernel evaluations, and the rescaled last step costs $\widetilde{O}(\sqrt{n\beta s}\,(s\mathcal{T}_\mathsf{K} + s^2))$ additional time and $\widetilde{O}(\sqrt{n\beta s} \cdot s)$ additional kernel evaluations.*

*Proof.* As in the proof of Theorem C.25 and in QLS of Algorithm 4, Algorithm 9 runs Algorithm 2 on the rows of $\Phi$ with the weights of Lemma C.24 at rank $k$ and constant accuracy at every level, followed by one additional oracle level. The chain $S_0 \subset_{1/2} \cdots \subset_{1/2} S_T = [n]$ is the chain of Algorithm 2 with $\mathbb{E}\,|S_0| = n2^{-T} \le s$, and $C_{t-1} := D_{t-1}^\top \Phi$ is the weighted sample of level $t-1$, with $C_0 = \Phi_{S_0}$. We never form the rows of $\Phi$, since every quantity the algorithm uses is an inner product $\phi_i^\top \phi_j = \mathsf{K}(x_i, x_j)$.

**Algorithm 9** Quantum generalized ridge leverage score sampling via recursive Nyström method.

**procedure** QNYSTRÖMKERNEL($\{x_1, \dots, x_n\} \in (\mathbb{R}^d)^n, \mathsf{K} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}, s, \delta$)
  $c \leftarrow 100$
  $T \leftarrow \max\{1, \lceil \log_2(n/s) \rceil\}$, $\delta' \leftarrow \delta/(2T)$ ▷ $\delta'$ is the failure parameter of each level of the recursion.
  Let $S_0 \subset_{1/2} S_1 \subset_{1/2} \dots \subset_{1/2} S_T = [n]$ ▷ Starting from $[n]$, uniformly sampling half of the indices, and $\mathbb{E}\,|S_0| \le s$.
  Set $k$ to be the largest integer with $ck \log(2k/\delta) \le s$
  Let $D_0 \in \mathbb{R}^{n \times |S_0|}$ be the sampling matrix of $S_0$ (weights 1)
  **for** $t = 1 \to T + 1$ **do**
    $M_{t-1} \leftarrow \{D_{t-1}(i) D_{t-1}(j) \cdot \mathsf{K}(x_i, x_j)\}_{(i,j) \in D_{t-1} \times D_{t-1}}$ ▷ $M_{t-1} = C_{t-1} C_{t-1}^\top$ for $C_{t-1} = D_{t-1}^\top K^{1/2}$, of size $\widetilde{O}(s) \times \widetilde{O}(s)$.
    $\lambda \leftarrow \frac{1}{k} \sum_{i>k} \sigma_i(M_{t-1})$ ▷ $\lambda = \|C_{t-1} - (C_{t-1})_k\|_F^2/k$.
    $\widehat{M} \leftarrow (M_{t-1} + \lambda I)^{-1}$
    $S \leftarrow S_{\min\{t,T\}}$ ▷ $S = [n]$ at the additional level $t = T + 1$.
    ▷ $D_{t-1}^\top K_i := \{D_{t-1}(j)\, \mathsf{K}(x_i, x_j)\}_{j \in D_{t-1}}$ for $i \in S$, where $D_{t-1}(j)$ = weight of $x_j$ in $D_{t-1}$
    Implement oracle for $q_i \leftarrow \frac{5}{\lambda} \cdot (\mathsf{K}(x_i, x_i) - (D_{t-1}^\top K_i)^\top \widehat{M} D_{t-1}^\top K_i)$ for $i \in S$ ▷ $q_i = 5\, w_i(\Phi_S, C_{t-1})$ for $\Phi = K^{1/2}$, by the Woodbury identity (Lemma E.1).
    ▷ $p_i = \min\{1, 16 q_i \log(2k/\delta')\}$ for $t \le T$, and $p_i = \min\{1, 16 q_i \log(2k/\delta)\}$ at $t = T + 1$ (as in Algorithm 4)
    $\widetilde{D}_t \leftarrow$ QSAMPLE$(p, \delta/(2(T+2)))$ ▷ $\widetilde{D}_t \in \mathbb{R}^{|S| \times \widetilde{O}(s)}$.
    $D_t \leftarrow D_S \cdot \widetilde{D}_t$ ▷ $D_t \in \mathbb{R}^{n \times \widetilde{O}(s)}$, weights $p_i^{-1/2}$.
  **end for**
  **return** $D_{T+1}$ and the oracle $q$ of the last level
**end procedure**

*The oracle.* Note that $M_{t-1} = D_{t-1}^\top K D_{t-1} = C_{t-1} C_{t-1}^\top$, so the value $\lambda$ computed at level $t$ is $\frac{1}{k} \sum_{i>k} \sigma_i(C_{t-1})^2 = \lambda_{(C_{t-1})_k}$. For $\lambda > 0$, the Woodbury identity gives

$$(C_{t-1}^\top C_{t-1} + \lambda I)^{-1} = \frac{1}{\lambda}\Big(I - \Phi^\top D_{t-1} (M_{t-1} + \lambda I)^{-1} D_{t-1}^\top \Phi\Big).$$

Since $\Phi \phi_i = K_i$ is the $i$-th column of $K$, the weight in Lemma C.24 of a row $\phi_i$ with $i \in S$ is

$$\begin{aligned} w_i(\Phi_S, C_{t-1}) &= \phi_i^\top (C_{t-1}^\top C_{t-1} + \lambda I)^{-1} \phi_i \\ &= \frac{1}{\lambda}\Big(\mathsf{K}(x_i, x_i) - (D_{t-1}^\top K_i)^\top (M_{t-1} + \lambda I)^{-1} D_{t-1}^\top K_i\Big), \end{aligned}$$

so the oracle of Algorithm 9 returns $q_i = 5\, w_i(\Phi_S, C_{t-1})$ exactly. Once $\widehat{M}$ is formed, each query takes $\widetilde{O}(s)$ kernel evaluations and $\widetilde{O}(s^2)$ time. If $\lambda = 0$, i.e., $\operatorname{rank}(M_{t-1}) \le k$, the oracle follows the two cases of Algorithm 6, which we omit from the display. It returns $\infty$ if $\mathsf{K}(x_i, x_i) > (D_{t-1}^\top K_i)^\top M_{t-1}^\dagger D_{t-1}^\top K_i$, i.e., $\phi_i \notin \operatorname{span}(C_{t-1}^\top C_{t-1})$, and $5\,(D_{t-1}^\top K_i)^\top (M_{t-1}^\dagger)^2 D_{t-1}^\top K_i = 5\, \phi_i^\top (C_{t-1}^\top C_{t-1})^\dagger \phi_i$ otherwise.

*The recursion.* As in the proof of Theorem C.25, we use Lemma C.22 at accuracy $0.01$ and failure probability $\delta' = \delta/(2T)$ at each of the $T$ levels in place of items 2 and 3 of Assumption C.1. Since the oracle is exact, the factor $1 \pm 0.04$ of Lemma C.21 is absorbed by the literal constant $5$. Moreover, the oversampling of Lemma C.22 is absorbed by the constant $16$, which, like $c$, stands for a sufficiently large constant. Hence, with probability at least $1 - T\delta' = 1 - \delta/2$, every $C_t$ is a $0.01$-approximator of $\Phi_{S_t}$ of expected size $O(k \log(k/\delta')) = O(s + k \log\log n) = \widetilde{O}(s)$, and $C_T$ is a $0.01$-approximator of $\Phi$.

*The additional level.* By Lemma C.21 with $\epsilon = 0.01$ and $C = \Phi$, we have $w_i(\Phi, C_T) = (1 \pm 0.04)\, w_i(\Phi, \Phi) = (1 \pm 0.04)\, \overline{\tau}_i(\Phi)$ for every $i$, and hence the last oracle returns $q_i = 5\, w_i(\Phi, C_T) \in [4.8\, \overline{\tau}_i(\Phi), 5.2\, \overline{\tau}_i(\Phi)]$. Thus, $\overline{\tau}_i(\Phi) \le q_i \le 6\, \overline{\tau}_i(\Phi)$ for every $i$, and $\sum_i q_i \le 5.2 \sum_i \overline{\tau}_i(\Phi) \le 10.4\, k \le 12k$ by Lemma C.24. The final sample is drawn by Lemma B.15

with $\sum_i p_i \le 16\log(2k/\delta)\sum_i q_i = O(k\log(k/\delta))$, and with the rescaling this bound becomes $O(\beta k\log(n/\delta))$. The sampler is called at most $T+2$ times, for $S_0$, the $T$ levels, and the additional level, each time with failure probability $\delta/(2(T+2))$, so the failure probability is at most $\delta/2+\delta/2=\delta$.

*Running time.* The bounds follow from Theorem C.3 as in the proof of Theorem C.25, using the oracle costs computed above. Each of the $T+1=O(\log n)$ oracle preparations forms $M_{t-1}$ from $\widetilde{O}(s^2)$ kernel evaluations, and computes its eigendecomposition and $\widehat{M}$ in $\widetilde{O}(s^\omega)$ time. Each of the $T+2$ sampler calls makes $\widetilde{O}(\sqrt{ns})$ queries in expectation, except that the rescaled last call makes $\widetilde{O}(\sqrt{n\beta s})$ queries in expectation. Each query costs $\widetilde{O}(s)$ kernel evaluations and $\widetilde{O}(s^2)$ time. Here the expectations are understood as in Theorem C.3. □

**Lemma E.2** (Ridge leverage scores of the rows of $KD$)**.** *Let $K\in\mathbb{R}^{n\times n}$ be positive semidefinite with $\operatorname{rank}(K)>m$, let $\rho_i:=\overline{\tau}_i(K^{1/2})$ be the rank-$m$ ridge leverage scores of the rows of $K^{1/2}$, $\lambda:=\frac{1}{m}\sum_{j>m}\sigma_j(K)=\lambda_{(K^{1/2})_m}$ and $\lambda':=\|K-K_m\|_F^2/m$. Let $D\in\mathbb{R}^{n\times t}$ select column $i$ of $K$ independently with probability $p_i\ge\min\{1,\beta\sqrt{n/m}\,\rho_i\}$ and weight $p_i^{-1/2}$, where $\beta\ge c\log(n/\delta)$ for a sufficiently large constant $c$, let $C:=KD$, and let $c_i^\top$ be the $i$-th row of $C$. Then with probability at least $1-\delta$, for every $i\in[n]$ and every $\mu\ge 0.9\lambda'$,*

$$c_i^\top(C^\top C+\mu I)^{-1}c_i\le 3\sqrt{n/m}\cdot\rho_i.$$

*Proof.* Let $\Psi:=K^{1/2}D$. Then $C=K^{1/2}\Psi$, $C^\top C=\Psi^\top K\Psi$, $c_i=\Psi^\top\psi_i$, and $\rho_i=\psi_i^\top(K+\lambda I)^{-1}\psi_i$ with $\psi_i:=K^{1/2}e_i$. Let $\lambda_s:=\lambda\sqrt{m/n}$ and $\epsilon_1:=1/10$. We will use two facts about the spectrum. First, the Cauchy–Schwarz inequality $(\sum_{j>m}\sigma_j(K))^2\le(n-m)\sum_{j>m}\sigma_j(K)^2$ gives $\lambda^2\le\frac{n-m}{m}\lambda'\le\frac{n}{m}\lambda'$, and hence $\lambda_s^2\le\lambda'$. Second, for every $x\ge 0$, we have

$$\frac{x}{x^2+\lambda'}\le\frac{2\sqrt{n/m}}{x+\lambda},\tag{3}$$

because $x(x+\lambda)\le 2x^2$ if $x\ge\lambda$, and if $x<\lambda$, then $x(x+\lambda)\le 2\lambda x\le(\lambda/\sqrt{\lambda'})(x^2+\lambda')\le\sqrt{n/m}\,(x^2+\lambda')$ by $x^2+\lambda'\ge 2x\sqrt{\lambda'}$.

*The sample $\Psi$ approximates $K$ at the ridge parameter $\lambda_s$.* Note that $(K+\lambda_s I)^{-1}\preceq(\lambda/\lambda_s)(K+\lambda I)^{-1}$, so the $\lambda_s$-ridge leverage scores $\tau_i:=\psi_i^\top(K+\lambda_s I)^{-1}\psi_i$ of the columns of $K^{1/2}$ satisfy $\tau_i\le(\lambda/\lambda_s)\rho_i=\sqrt{n/m}\,\rho_i$, and hence $\sum_i\tau_i\le\sqrt{n/m}\sum_i\rho_i\le 2\sqrt{nm}\le 2n$ by Lemma C.24. The proof of Lemma 21 of Cohen et al. [2017] is a matrix Bernstein bound in which the rank enters only through the trace of the variance proxy, which is the sum of the ridge leverage scores, and through the logarithm in the oversampling factor. We apply this proof to the columns of $K^{1/2}$ with the ridge parameter $\lambda_s$ in place of $\|A-A_k\|_F^2/k$, the overestimates $\sqrt{n/m}\,\rho_i\ge\tau_i$, the trace bound $2n$ in place of $2k$, and the oversampling factor $\beta\ge c\log(n/\delta)$, and obtain that with probability at least $1-\delta$,

$$(1-\epsilon_1)\Psi\Psi^\top-\epsilon_1\lambda_s I\preceq K\preceq(1+\epsilon_1)\Psi\Psi^\top+\epsilon_1\lambda_s I.\tag{4}$$

Assuming this event occurs, we show that the claim holds for $\mu=\lambda'$, and for $\mu\ge 0.9\lambda'$ the claim then follows from $(C^\top C+\mu I)^{-1}\preceq\frac{1}{0.9}(C^\top C+\lambda' I)^{-1}$.

*From $\Psi^\top K\Psi$ to $\Psi^\top\Psi$.* Let $T:=\Psi^\top\Psi\in\mathbb{R}^{t\times t}$ and $G:=\Psi\Psi^\top\in\mathbb{R}^{n\times n}$. Conjugating the left inequality of (4) by $\Psi$ gives $\Psi^\top K\Psi\succeq(1-\epsilon_1)T^2-\epsilon_1\lambda_s T$, and since $2\lambda_s\nu\le\nu^2+\lambda_s^2\le\nu^2+\lambda'$ for every eigenvalue $\nu$ of $T$, we have

$$C^\top C+\lambda' I=\Psi^\top K\Psi+\lambda' I\succeq(1-\epsilon_1)T^2-\tfrac{\epsilon_1}{2}(T^2+\lambda' I)+\lambda' I\succeq(1-\tfrac{3\epsilon_1}{2})(T^2+\lambda' I).$$

Hence, $c_i^\top(C^\top C+\lambda' I)^{-1}c_i\le(1-\frac{3\epsilon_1}{2})^{-1}\,\psi_i^\top\Psi(T^2+\lambda' I)^{-1}\Psi^\top\psi_i=(1-\frac{3\epsilon_1}{2})^{-1}\,\psi_i^\top G(G^2+\lambda' I)^{-1}\psi_i$. The last step uses the identity $\Psi(T^2+\lambda' I)^{-1}\Psi^\top=G(G^2+\lambda' I)^{-1}$, which holds since both sides equal $U\Sigma^2(\Sigma^4+\lambda' I)^{-1}U^\top$ for a compact singular value decomposition $\Psi=U\Sigma V^\top$.

*From $G$ to $K$.* Applying (3) to the eigenvalues of $G$ gives $G(G^2+\lambda' I)^{-1}\preceq 2\sqrt{n/m}\,(G+\lambda I)^{-1}$, and since $\lambda_s\le\lambda$, the right inequality of (4) gives $K+\lambda I\preceq(1+\epsilon_1)G+(\lambda+\epsilon_1\lambda_s)I\preceq(1+\epsilon_1)(G+\lambda I)$,

i.e., $(G+\lambda I)^{-1} \preceq (1+\epsilon_1)(K+\lambda I)^{-1}$. Putting everything together, we have

$$c_i^\top (C^\top C + \lambda' I)^{-1} c_i \le \frac{2(1+\epsilon_1)}{1-3\epsilon_1/2}\sqrt{n/m}\cdot \psi_i^\top (K+\lambda I)^{-1}\psi_i \le 2.6\sqrt{n/m}\cdot \rho_i,$$

and $2.6/0.9 \le 3$ gives the claim for $\mu \ge 0.9\lambda'$. □

**Theorem E.3.** *There exists a quantum algorithm (Algorithm 8, run with $\delta = 0.001$) that given any set of points $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$ and a positive definite kernel function $\mathsf{K} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and any positive integer $k \le n$ and $\epsilon \in (0,1)$ with $k/\epsilon \le n$ (we write $m := k/\epsilon$ and treat it as an integer, and otherwise replace it by $\lceil k/\epsilon \rceil$ throughout), runs in expected*

$$\widetilde{O}\big(n^{0.75}k^{1.25}\epsilon^{-1.25}(\mathcal{T}_\mathsf{K} + k\epsilon^{-1}) + n^{0.5}k^{1.5}\epsilon^{-2.5}(\mathcal{T}_\mathsf{K}+1) \\ + n^{0.5}k^{\omega-0.5}\epsilon^{0.5-\omega} + \mathrm{poly}(k/\epsilon)\cdot(\mathcal{T}_\mathsf{K}+1)\big)$$

*time, where $\mathcal{T}_\mathsf{K}$ is the time to evaluate $\mathsf{K}$ on any pair of points $x_i, x_j$, and makes, in expectation, $\widetilde{O}(\epsilon^{-2.5}n^{0.5}k^{1.5} + \epsilon^{-1.25}n^{0.75}k^{1.25} + \mathrm{poly}(k/\epsilon))$ queries to the points in $X$, and returns a pair of rank-$k$ matrices $M, N \in \mathbb{R}^{n\times k}$ (given implicitly in factor form) such that*

$$\|K - MN^\top\|_F^2 \le (1+\epsilon)\|K - K_k\|_F^2$$

*holds with probability at least $0.99$.*

**Remark E.4** (PSD low-rank approximation)**.** *A positive semidefinite matrix $K \in \mathbb{R}^{n\times n}$ given by entry access is the kernel matrix of the linear kernel on the rows of any factor $\Phi$ with $K = \Phi\Phi^\top$, and one entry read is one kernel evaluation. Since the algorithm accesses the points only through kernel evaluations, $\Phi$ is never formed, and hence, for $k$ and $\epsilon$ as in Theorem E.3, the theorem applies with $\mathcal{T}_\mathsf{K} = O(1)$. The expected time is then $\widetilde{O}(n^{0.75}k^{2.25}\epsilon^{-2.25} + n^{0.5}k^{1.5}\epsilon^{-2.5} + n^{0.5}k^{\omega-0.5}\epsilon^{0.5-\omega} + \mathrm{poly}(k/\epsilon))$. Moreover, the proof of Theorem E.3 counts the kernel evaluations, which here are entry reads, and gives $\widetilde{O}(\epsilon^{-1.25}n^{0.75}k^{1.25} + \epsilon^{-2.5}n^{0.5}k^{1.5} + \mathrm{poly}(k/\epsilon))$ of them in expectation. The leading terms are the PSD rows of Tables 2 and 4.*

*Proof.* Our algorithm could be interpreted as a quantum implementation of a generalization of Bakshi et al. [2020], where they only tackle the case $\mathsf{K}(x_i, x_j) = x_i^\top x_j$, in which the kernel matrix $K$ is given directly. Throughout the proof, we write $\mathrm{OPT}_k := \|K - K_k\|_F^2$, and we assume without loss of generality that $\epsilon \le 1/2$, since $\epsilon$ is rescaled by a constant at the end of the proof.

**What is imported from Bakshi et al. [2020].** Algorithm 8 follows Algorithm 3 of Bakshi et al. [2020], which samples a column sketch $C$ and a row sketch $R$ of $K$, computes an orthonormal $Z$ that spans a rank-$m$ spectral low-rank approximation of $R$, solves the spectral regression $\min_W \|C - WZ^\top\|_2$ by their Algorithm 2, and applies their Algorithm 1 to the column span of $W$. The proof of their Theorem 4.16 for Algorithm 1 uses only the following two properties of the orthonormal basis $Q$ it receives: (Q1) $QQ^\top$ is an $(\epsilon, k)$-spectral-Frobenius projection for $K$, i.e., $\|K - QQ^\top K\|_2^2 \le (\epsilon/k)\cdot \mathrm{OPT}_k$, which enters only once, through Lemma 4.14 of Bakshi et al. [2020] (due to Clarkson and Woodruff [2017b]), to give

$$\min_{X:\mathrm{rank}(X)\le k} \|K - QXQ^\top\|_F^2 \le \|K - QQ^\top K_k QQ^\top\|_F^2 \le (1+\epsilon)\cdot \mathrm{OPT}_k, \tag{5}$$

and (Q2) the leverage scores of $Q$ are its squared row norms, which are used to sample rows and columns of $K$ for the rank-constrained regression and for the final regression. In our algorithm, the column span of $W$ plays the role of $Q$, the leverage scores of $W$ are the squared row norms of $Q$, and we sample by them with QLS instead of reading them off $Q$. Therefore, the proof has two parts, where Part 1 establishes (Q1) for an orthonormal basis $Q$ of the column span of $W$, and Part 2 solves the regressions in the basis $W$ from sampled rows and columns.

**Part 1: the column span of $W$.** We use the following two properties of $D_1$ and $D_2$ at rank $m$ and accuracy $0.1$. They are the projection-cost preservation statements behind steps 2 and 3 of Algorithm 3 of Bakshi et al. [2020], which are their Lemmas 4.11 and 4.12, due to Musco and Woodruff [2017]. Bakshi et al. [2020] draw $t$ columns and $t$ rows with replacement by constant-factor approximations of the rank-$m$ ridge leverage scores of $K^{1/2}$, computed by the Nyström method of Musco and Musco

[2017] in linear time. Here $D_1$ and $D_2$ include every index independently, with the probabilities of Lemma E.1 rescaled by $\sqrt{\epsilon n/k} = \sqrt{n/m}$. Throughout, the numerical constants $c$, 16, and 100 of Algorithms 8 and 9 stand for sufficiently large absolute constants, chosen such that each sampling lemma we invoke applies at the stated accuracy with failure probability at most $0.001$. They enter the bounds only through the constants of the $O(\cdot)$ notation, whereas the constant 5 is literal, since the bounds $q_i \le 6\rho_i$ and $\sum_i q_i \le 12m$ below use it.

(P1) For every orthogonal projection $X \in \mathbb{R}^{n\times n}$ of rank at most $m$: $\|C - XC\|_F^2 = (1 \pm 0.1)\|K - XK\|_F^2$ and

$$\tfrac{9}{10}\|K - XK\|_2^2 - \tfrac{1}{10m}\|K - K_m\|_F^2 \le \|C - XC\|_2^2 \le \tfrac{11}{10}\|K - XK\|_2^2 + \tfrac{1}{10m}\|K - K_m\|_F^2.$$

(P2) For every orthogonal projection $X \in \mathbb{R}^{t\times t}$ of rank at most $m$: $\|R - RX\|_F^2 = (1 \pm 0.1)\|C - CX\|_F^2$ and

$$\tfrac{9}{10}\|C - CX\|_2^2 - \tfrac{1}{10m}\|C - C_m\|_F^2 \le \|R - RX\|_2^2 \le \tfrac{11}{10}\|C - CX\|_2^2 + \tfrac{1}{10m}\|C - C_m\|_F^2.$$

*Proof of (P1) and (P2).* Let $\rho_i := \overline{\tau}_i(K^{1/2})$, $\lambda := \frac{1}{m}\sum_{j>m}\sigma_j(K)$, and $\lambda' := \|K - K_m\|_F^2/m$ be as in Lemma E.2, and assume $\mathrm{rank}(K) > m$. Otherwise $K = K_m$ and $\lambda = \lambda' = 0$, so $\overline{\tau}_i(K) = (P_K)_{ii} = \rho_i$ for the orthogonal projector $P_K$ onto $\mathrm{range}(K)$, and, for the column sample $C = KD_1$, the score $\overline{\tau}_i(C^\top)$ is the $i$-th diagonal entry of the projector onto $\mathrm{range}(C) \subseteq \mathrm{range}(K)$, hence it is at most $\rho_i$. Thus, the overestimates $\widetilde{\tau}_i$ and $\widehat{\tau}_i$ used below remain valid also when $\mathrm{rank}(K) \le m$, with the pseudo-inverse convention of Section C.3. We apply Lemma E.1 with $s = k'$, which makes its rank parameter $m$, and with $\beta = \sqrt{\epsilon n/k} = \sqrt{n/m}$. If both calls succeed, then $D_1$ and $D_2$ include index $i$ independently with probability $p_i = \min\{1, \sqrt{n/m}\cdot 16\log(12n/\delta)\, q_i\}$ and weight $p_i^{-1/2}$, where $\rho_i \le q_i \le 6\rho_i$, and hence $\sum_i q_i \le 12m$ and $\sum_i p_i = O(\sqrt{nm}\log(n/\delta)) = O(t)$. We use two facts from Bakshi et al. [2020] and Cohen et al. [2017]. First, the rank-$m$ ridge leverage scores of the columns of $K$ satisfy $\overline{\tau}_i(K) \le 2\sqrt{n/m}\,\rho_i$, which is Lemma 4.10 of Bakshi et al. [2020], due to Musco and Woodruff [2017]. In the eigenbasis $K = \sum_j \sigma_j v_j v_j^\top$, these two scores are $\sum_j v_{ji}^2\sigma_j^2/(\sigma_j^2 + \lambda')$ and $\sum_j v_{ji}^2\sigma_j/(\sigma_j + \lambda)$, and the ratio of the $j$-th terms is at most 2 if $\sigma_j \ge \lambda$. In the other case, the ratio is at most $\lambda/\sqrt{\lambda'}$ by $\sigma_j^2 + \lambda' \ge 2\sigma_j\sqrt{\lambda'}$, and $\lambda^2 \le \frac{n}{m}\lambda'$ by the Cauchy–Schwarz inequality. Second, let each column $a_i$ of a matrix $A$ be included independently with probability $p_i := \min\{1, c'\epsilon_1^{-2}\log(k/\delta)\,\widetilde{\tau}_i\}$ and weight $p_i^{-1/2}$, where $\widetilde{\tau}_i \ge \overline{\tau}_i(A)$ are overestimates of the rank-$k$ ridge leverage scores of the columns and $c'$ is a sufficiently large constant. Then by Lemma 21 of Cohen et al. [2017], with probability at least $1 - \delta$, the sample $B$ has $O(\epsilon_1^{-2}\log(k/\delta)\sum_i \widetilde{\tau}_i)$ columns and satisfies $(1-\epsilon_1)BB^\top - \epsilon_1\lambda_{A_k} I \preceq AA^\top \preceq (1+\epsilon_1)BB^\top + \epsilon_1\lambda_{A_k} I$. On the same event, it also satisfies $\|B - XB\|_F^2 = (1 \pm \epsilon_1)\|A - XA\|_F^2$ for every orthogonal projection $X$ of rank at most $k$ by their Theorem 6, whose proof uses $X$ only through $\mathrm{rank}(X) \le k$ and, as they remark after it, applies to independent sampling with their Lemmas 21 and 22 in place of Theorem 5 and Lemma 20. For $\epsilon_1 = 1/11$, conjugating the spectral inequality by $I - X$ and taking spectral norms gives

$$\tfrac{9}{10}\|A - XA\|_2^2 - \tfrac{1}{10}\lambda_{A_k} \le \|B - XB\|_2^2 \le \tfrac{11}{10}\|A - XA\|_2^2 + \tfrac{1}{10}\lambda_{A_k},$$

since $\frac{1}{1-\epsilon_1} = \frac{11}{10}$, $\frac{\epsilon_1}{1-\epsilon_1} = \frac{1}{10}$, $\frac{1}{1+\epsilon_1} \ge \frac{9}{10}$, and $\frac{\epsilon_1}{1+\epsilon_1} \le \frac{1}{10}$. (P1) is the case $A = K$, $B = C = KD_1$, $k = m$, and $\lambda_{A_m} = \frac{1}{m}\|K - K_m\|_F^2$, with the overestimates $\widetilde{\tau}_i := 2\sqrt{n/m}\,q_i \ge \overline{\tau}_i(K)$, for which $p_i = \min\{1, 8\log(12n/\delta)\,\widetilde{\tau}_i\}$ and $\sum_i \widetilde{\tau}_i \le 24\sqrt{nm}$, so the sample has $O(t)$ columns. (P2) is the case $A = C^\top$, whose columns are the rows $c_i$ of $C$, $B = R^\top = (D_2^\top C)^\top$, $k = m$, and $\lambda_{A_m} = \frac{1}{m}\|C - C_m\|_F^2 =: \lambda_C$, applied conditionally on $D_1$. Note that we may condition on $D_1$ since $D_2$ is independent of $D_1$. The overestimates are $\widehat{\tau}_i := 3\sqrt{n/m}\,q_i$, where $q_i$ now denotes the last-oracle values of the second call, which also lie in $[\rho_i, 6\rho_i]$. The Frobenius part of (P1), applied to the projection $X$ onto the top-$m$ left singular vectors of $C$, gives $\|C - C_m\|_F^2 \ge \frac{9}{10}\|K - XK\|_F^2 \ge \frac{9}{10}\|K - K_m\|_F^2$, i.e., $\lambda_C \ge 0.9\lambda'$. Hence, by Lemma E.2 applied to $D_1$ with $\beta = 16\log(12n/\delta)$,

with probability at least 0.999 over $D_1$ we have $\overline{\tau}_i(C^\top) = c_i^\top (C^\top C + \lambda_C I)^{-1} c_i \le 3\sqrt{n/m}\,\rho_i \le \widehat{\tau}_i$ for every $i$. Then $p_i = \min\{1, \frac{16}{3}\log(12n/\delta)\,\widehat{\tau}_i\}$ and $\sum_i \widehat{\tau}_i \le 36\sqrt{nm}$, so the sample has $O(t)$ rows.

Minimizing the Frobenius parts over $X$ gives $\|C - C_m\|_F^2 \le 1.1\|K - K_m\|_F^2 \le 1.1\ \mathrm{OPT}_k$ and $\|R - R_m\|_F^2 \le 1.1\|C - C_m\|_F^2 \le 1.21\ \mathrm{OPT}_k$.

*The matrix $Z$.* Corollary D.3 is Theorem C.25 applied to the columns of $R^\top = D_1^\top K D_2 \in \mathbb{R}^{t\times t}$, and hence the output of the corollary is an $\epsilon_0$-approximator. That is, $\widetilde{R} \in \mathbb{R}^{t\times \widetilde{O}(k/\epsilon)}$ is a weighted subset of the columns of $R^\top$ which is an $\epsilon_0$-coreset of them for the $m$-subspace cost and satisfies

$$(1-\epsilon_0)\widetilde{R}\widetilde{R}^\top - \epsilon_0\lambda I \preceq R^\top R \preceq (1+\epsilon_0)\widetilde{R}\widetilde{R}^\top + \epsilon_0\lambda I, \qquad \lambda := \|R - R_m\|_F^2/m.$$

The matrix $Z \in \mathbb{R}^{t\times m}$ consists of the top-$m$ left singular vectors of $\widetilde{R}$. Its rows are indexed by the samples of $D_1$, as are the columns of $C$, so $C - WZ^\top$ is well defined. We derive two consequences. First, by Lemma 3 of Cohen et al. [2015a] as stated in Section D, with $\gamma = \epsilon_0$ in the remark following it, we have

$$\|R - RZZ^\top\|_F^2 = \|(I - ZZ^\top)R^\top\|_F^2 \le (1+3\epsilon_0)\|R - R_m\|_F^2. \tag{6}$$

Second, a unit vector $v$ orthogonal to the columns of $Z$ satisfies $v^\top \widetilde{R}\widetilde{R}^\top v \le \sigma_{m+1}(\widetilde{R})^2$, and the upper spectral inequality gives $v^\top R^\top R v \le (1+\epsilon_0)\sigma_{m+1}(\widetilde{R})^2 + \epsilon_0\lambda$. Moreover, the lower spectral inequality gives $(1-\epsilon_0)\sigma_{m+1}(\widetilde{R})^2 \le \sigma_{m+1}(R)^2 + \epsilon_0\lambda$ by the monotonicity of eigenvalues under the positive semidefinite order. Since $\|R - RZZ^\top\|_2^2 = \max\{v^\top R^\top R v : \|v\|_2 = 1, Z^\top v = 0\}$, we have

$$\|R - RZZ^\top\|_2^2 \le \frac{1+\epsilon_0}{1-\epsilon_0}\big(\sigma_{m+1}(R)^2 + \epsilon_0\lambda\big) + \epsilon_0\lambda \le 1.03\,\sigma_{m+1}(R)^2 + 0.03\,\lambda. \tag{7}$$

To bound $\sigma_{m+1}(R)^2$, we use the upper inequalities of (P2) and (P1). Let $X_C$ be the projection onto the top-$m$ right singular vectors of $C$ and $X_K$ the projection onto the top-$m$ left singular vectors of $K$. Then we have

$$\begin{aligned}
\sigma_{m+1}(R)^2 &\le \|R - RX_C\|_2^2 \le \tfrac{11}{10}\sigma_{m+1}(C)^2 + \tfrac{1}{10m}\|C - C_m\|_F^2,\\
\sigma_{m+1}(C)^2 &\le \|C - X_K C\|_2^2 \le \tfrac{11}{10}\sigma_{m+1}(K)^2 + \tfrac{1}{10m}\|K - K_m\|_F^2,
\end{aligned}$$

and $\sigma_{m+1}(K)^2 \le \frac{1}{m-k}\sum_{i=k+1}^{m+1}\sigma_i(K)^2 \le \frac{2\epsilon}{k}\,\mathrm{OPT}_k$ because $m \ge 2k$. Combining these with the Frobenius tail bounds above gives $\sigma_{m+1}(C)^2 \le 2.3\frac{\epsilon}{k}\,\mathrm{OPT}_k$, $\sigma_{m+1}(R)^2 \le 2.7\frac{\epsilon}{k}\,\mathrm{OPT}_k$, and $\lambda \le 1.21\frac{\epsilon}{k}\,\mathrm{OPT}_k$. Therefore, by (7) and (6),

$$\|R - RZZ^\top\|_2^2 \le 2.9\,\frac{\epsilon}{k}\,\mathrm{OPT}_k, \qquad \|R - RZZ^\top\|_F^2 \le 1.3\ \mathrm{OPT}_k, \tag{8}$$

and by the lower inequalities of (P2) with $X = ZZ^\top$,

$$\begin{aligned}
\|C - CZZ^\top\|_2^2 &\le \tfrac{10}{9}\Big(\|R - RZZ^\top\|_2^2 + \tfrac{1}{10m}\|C - C_m\|_F^2\Big) \le 3.4\,\frac{\epsilon}{k}\,\mathrm{OPT}_k,\\
\|C - CZZ^\top\|_F^2 &\le \tfrac{10}{9}\|R - RZZ^\top\|_F^2 \le 1.5\ \mathrm{OPT}_k\,.
\end{aligned} \tag{9}$$

*The matrix $W$.* The matrix $D_3$ includes the $i$-th row of $Z$ independently with probability $p_i = \min\{1, \alpha\|z_i\|_2^2\}$, where $\alpha := \epsilon k'/k = 100\log(12k/(\epsilon\delta))$, and rescales it by $p_i^{-1/2}$. This is the sampling matrix $S$ of Algorithm 2 of Bakshi et al. [2020], which samples by the leverage scores of $Z^\top$ oversampled by a logarithmic factor, and these leverage scores are the squared row norms of $Z$. Moreover, $W = CD_3(Z^\top D_3)^\dagger$ is its output $\widehat{W} = CS(Z^\top S)^\dagger$. The proof of Theorem 4.18 of Bakshi et al. [2020], via their Lemma 4.19, equations (4.18)–(4.21), and Lemma 4.25, uses only that $Z$ has orthonormal columns and that $S$ is drawn in this way. It shows that, with probability at least 0.999,

$$\|C - WZ^\top\|_2^2 \le \widetilde{O}(1)\cdot\Big(\|C - CZZ^\top\|_2^2 + \frac{1}{m}\|C - CZZ^\top\|_F^2\Big) \le \widetilde{O}(1)\cdot\frac{\epsilon}{k}\,\mathrm{OPT}_k, \tag{10}$$

where the second step follows from (9). Their statement assumes in addition that $ZZ^\top$ is an $(O(1), m)$-spectral-Frobenius projection for $C$, but this assumption serves only to rewrite the first

bound in terms of $\|C - C_m\|_F^2$, and we use the first bound directly. The event of (10) is the matrix Bernstein bound of their Lemma 4.22, which is Lemma 21 of Cohen et al. [2017], together with Markov bounds on expectations. The failure probability of the matrix Bernstein bound enters through the oversampling factor $\alpha \geq 100 \log(1000m)$, so a failure probability of $0.001$ costs a constant factor in the $\widetilde{O}(1)$.

Let $Q$ be an orthonormal basis of the column span of $W$, a span that contains $WZ^\top y$ for every $y$, so $\|(I - QQ^\top)Cy\|_2 \leq \|Cy - WZ^\top y\|_2$ and hence $\|C - QQ^\top C\|_2^2 \leq \|C - WZ^\top\|_2^2$. Applying the lower inequality of (P1) to $X = QQ^\top$, whose rank is at most $m$, we have

$$\|K - QQ^\top K\|_2^2 \leq \tfrac{10}{9}\Big(\|C - QQ^\top C\|_2^2 + \tfrac{1}{10m}\|K - K_m\|_F^2\Big) \leq \widetilde{O}(1) \cdot \frac{\epsilon}{k}\,\mathrm{OPT}_k\,. \tag{11}$$

This is (Q1) with $\epsilon$ replaced by $\epsilon \cdot \mathrm{polylog}(n, k/\epsilon)$, so we run the algorithm with $\epsilon/\mathrm{polylog}(n, k/\epsilon)$ in place of $\epsilon$, and the $\widetilde{O}(\cdot)$ in the running time absorbs this change. Thus, we obtain $\|K - QQ^\top K\|_2^2 \leq (\epsilon/k) \cdot \mathrm{OPT}_k$ and hence (5).

**Part 2: the rank-constrained regression in the basis $W$.** We will focus on solving the rank-constrained regression $\min_{Y:\mathrm{rank}(Y)\leq k} \|K - WYW^\top\|_F^2$, which is the major divergence between our approach and that of Bakshi et al. [2020]. In Bakshi et al. [2020], since they could afford linear in $n$ time, they compute an orthonormal basis for $W$ denoted by $Q$, and instead solve the regression $\min_{X:\mathrm{rank}(X)\leq k} \|K - QXQ^\top\|_F^2$. Let $Q$ be the orthonormal basis of the column span of $W$ from Part 1, and let $T \in \mathbb{R}^{\mathrm{rank}(W)\times k/\epsilon}$ be the change-of-basis matrix such that $QT = W$. The matrix $T$ has full row rank, so $WT^\dagger = QTT^\dagger = Q$, and we observe that $X$ could be recovered via the following procedures:

- Solve
$$\min_{Y:\mathrm{rank}(Y)\leq k} \|K - WYW^\top\|_F^2 \tag{12}$$
and let $Y'$ denote the optimal solution of the above regression.
- Set $X_* := TY'T^\top$.

To see that $X_*$ is optimal for the rank-constrained regression against $Q$, note

$$\begin{aligned} QX_*Q^\top &= QTY'T^\top Q^\top \\ &= WY'W^\top, \end{aligned}$$

and if there exists a solution $X'$ with lower cost, then

$$\begin{aligned} \|K - WT^\dagger X'T^{\dagger\top}W^\top\|_F^2 &= \|K - QX'Q^\top\|_F^2 \\ &< \|K - QX_*Q^\top\|_F^2 \\ &= \|K - WY'W^\top\|_F^2, \end{aligned}$$

contradicting the definition of $Y'$. Both Bakshi et al. [2020] and Algorithm 8 construct leverage score sampling matrices according to the leverage scores of $W$ (in the context of Bakshi et al. [2020], they sample according to the row norms of $Q$, which are the leverage scores of $W$), then we solve the surrogate regression

$$\min_{Y:\mathrm{rank}(Y)\leq k} \|D_4^\top K D_5 - D_4^\top WYW^\top D_5\|_F^2, \tag{13}$$

so it suffices to show that the optimal solution of Eq. (13) is a good approximation to the optimal solution of Eq. (12). Let $f(Y) := \|K - WYW^\top\|_F^2$, $g(Y) := \|D_4^\top(K - WYW^\top)\|_F^2$, and $h(Y) := \|D_4^\top(K - WYW^\top)D_5\|_F^2$, and let $\mathrm{OPT} := \min_{Y:\mathrm{rank}(Y)\leq k} f(Y)$. By the change of basis above, OPT equals the left-hand side of (5), hence $\mathrm{OPT} \leq (1+\epsilon) \cdot \mathrm{OPT}_k$. The matrix $D_4$ samples $k'/\epsilon^2 = \Omega(\epsilon^{-2} m \log(m/\delta))$ rows by constant-factor approximations of the leverage scores of $W$, whose rank is at most $m$. The set $\{WYW^\top\}$ is not an affine space, but it is contained in $\{WH : H \in \mathbb{R}^{m\times n}\}$. Therefore, by Lemma B.12 with design $W$, response $K$, and failure probability $0.001$, together with Remark B.14, there is a constant $c_1$ with $g(Y) = (1 \pm \epsilon)f(Y) + c_1$ for all

$Y$, where $c_1 = \|D_4^\top K_*\|_F^2 - \|K_*\|_F^2$ with $K_* := K - WH_*$ and $H_* := \arg\min_H \|K - WH\|_F^2$. Similarly, $D_5$ samples $k'/\epsilon^2$ columns by the leverage scores of the rows of $W$. By Lemma B.12 with design $W$, response $(D_4^\top K)^\top$, and failure probability $0.001$, there is a constant $c_2$ with $\|(VW^\top - D_4^\top K)D_5\|_F^2 = (1 \pm \epsilon)\|VW^\top - D_4^\top K\|_F^2 + c_2$ for all $V \in \mathbb{R}^{k'/\epsilon^2 \times m}$, and taking $V = D_4^\top WY$ gives $h(Y) = (1 \pm \epsilon)g(Y) + c_2$ for all $Y$. Let $Y_*$ minimize $h$ over $\{Y : \operatorname{rank}(Y) \le k\}$, i.e., $Y_*$ solves Eq. (13), and recall that $Y'$ minimizes $f$ over the same set, so $f(Y') = \mathrm{OPT}$. Then $h(Y_*) \le h(Y')$, and $c_2$ cancels, so we have $(1-\epsilon)g(Y_*) \le (1+\epsilon)g(Y')$. Substituting $g = (1 \pm \epsilon)f + c_1$ on both sides, we obtain

$$(1-\epsilon)^2 f(Y_*) \le (1+\epsilon)^2 \ \mathrm{OPT} + 2\epsilon\,|c_1|. \tag{14}$$

The constant $c_1$ does not cancel, but it is small. To see this, we first observe that $\|K_*\|_F^2 = \min_H \|K - WH\|_F^2 \le \mathrm{OPT}$, because the unconstrained problem has the larger solution space. Moreover, $\|D_4^\top K_*\|_F^2$ is an unbiased estimator of $\|K_*\|_F^2$,

$$\mathbb{E}[\|D_4^\top K_*\|_F^2] = \mathbb{E}[\mathrm{tr}[K_*^\top D_4 D_4^\top K_*]] = \mathrm{tr}[K_*^\top \mathbb{E}[D_4 D_4^\top] K_*] = \mathrm{tr}[K_*^\top I_n K_*] = \|K_*\|_F^2,$$

since $D_4$ includes the $i$-th row with probability $q_i$ and rescales it by $q_i^{-1/2}$. By Markov's inequality, with probability at least $0.999$, we have $\|D_4^\top K_*\|_F^2 \le 1000\|K_*\|_F^2$, hence $|c_1| \le 999\|K_*\|_F^2 \le 999\ \mathrm{OPT}$ and, by (14),

$$f(Y_*) \le \frac{(1+\epsilon)^2 + 1998\epsilon}{(1-\epsilon)^2} \cdot \mathrm{OPT} = (1 + O(\epsilon)) \cdot \mathrm{OPT} \le (1 + O(\epsilon)) \cdot \mathrm{OPT}_k\,.$$

We apply the lemma of Friedland and Torokhti [2007] stated in Section D, which is Theorem 4.15 of Bakshi et al. [2020], with $B = D_4^\top W$, $C = W^\top D_5$, and $A = D_4^\top K D_5$ in its notation. It shows that the minimizer $Y_*$ of Eq. (13) admits the closed form computed in Algorithm 8.

We now have a rank-$k$ solution in factored form, which is $WY_*W^\top$ with $f(Y_*) \le (1+O(\epsilon)) \cdot \mathrm{OPT}_k$. Compute the top-$k$ left singular vectors of $Y_*$, denoted as $U_* \in \mathbb{R}^{m \times k}$, and write $Y_* = U_* V_*$ with $V_* := U_*^\top Y_*$, which is possible because $\operatorname{rank}(Y_*) \le k$. We set $M := WU_* \in \mathbb{R}^{n \times k}$ and plug the decomposition into the regression, so the right low-rank factor could be found by solving

$$\begin{aligned}\min_{N \in \mathbb{R}^{n \times k}} \|K - MN^\top\|_F^2 &\le \|K - WU_* V_* W^\top\|_F^2 \\ &\le (1 + O(\epsilon))\|K - K_k\|_F^2.\end{aligned}$$

To solve the regression, we employ leverage score sampling on the rows of $M$, whose rank is at most $k$. By Lemma B.13 with failure probability $0.001$, together with Remark B.14, it suffices to sample $O(k \log k + k/\epsilon)$ rows, and the solution $N_*^\top = (D_6^\top M)^\dagger (D_6^\top K)$ of the sketched regression satisfies

$$\begin{aligned}\|K - MN_*^\top\|_F^2 &\le (1+\epsilon) \min_{N \in \mathbb{R}^{n \times k}} \|K - MN^\top\|_F^2 \\ &\le (1 + O(\epsilon))\|K - K_k\|_F^2.\end{aligned}$$

Finally, by properly scaling $\epsilon$, we conclude the proof of correctness.

**Failure probability.** The six calls are QNYSTRÖMKERNEL for $D_1$ and $D_2$, QLOWRANKCMM for $\widetilde{R}$, and QLS for $D_4$, $D_5$, and $D_6$, and each call is made with failure parameter $\delta/6$, for a total of $\delta = 0.001$. Note that the property (P1), the event of Lemma E.2 for $D_1$, and the property (P2) each hold with probability at least $0.999$ for sufficiently large constants. The event (10) for $D_3$ has probability at least $0.999$. Moreover, the two applications of Lemma B.12, the application of Lemma B.13, the Markov step, and the sampler call for $D_3$ each fail with probability at most $0.001$. Therefore, by a union bound, the success probability is at least $1 - 0.001 - 0.003 - 0.001 - 0.005 = 0.99$.

Next, we analyze the runtime of Algorithm 8, item by item as follows:

- Forming the generalized ridge leverage score sampling matrices $D_1$ and $D_2$ (Algorithm 9) involves selecting $O(k'^2)$ entries from $K$, which could be implemented by $k'^2$ evaluations of the kernel function. In the loop, we compute the SVD of a $k' \times k'$ matrix, which takes $O(k'^\omega)$ time, and forming $\widehat{M}$ also takes $O(k'^\omega)$ time. Next, we need to analyze the complexity of

implementing the sampling oracle, for any fixed $i$, we form $D_{t-1}^\top K_i$ by forming a vector of length $k'$ through $k'$ kernel evaluations and an extra $k'^2$ time for the quadratic form. The last sampling step uses the probabilities $p_i = \min\{1, \sqrt{\epsilon n/k} \cdot 16 q_i \log(12n/\delta)\}$, whose sum is

$$\sum_{i=1}^{n} p_i = O(\sqrt{nk/\epsilon}\log(n/\delta)) = O(t)$$

by Lemma E.1. Therefore, by Lemma B.15, the last sampling step costs $\widetilde{O}(\sqrt{nt})\cdot(k'\mathcal{T}_\mathsf{K}+k'^2)$ time. Each of the $O(\log n)$ intermediate levels samples with $\sum_i p_i = \widetilde{O}(k')$ and costs $\widetilde{O}(\sqrt{nk'}) \cdot (k'\mathcal{T}_\mathsf{K} + k'^2) = \widetilde{O}(n^{0.5}k^{1.5}\epsilon^{-1.5}\mathcal{T}_\mathsf{K} + n^{0.5}k^{2.5}\epsilon^{-2.5})$. The first term of this cost is within the second term of the theorem, and the second term of this cost is at most $n^{0.75}k^{2.25}\epsilon^{-2.25}$ because $k/\epsilon \le n$, which is needed for $t \le n$. Thus, the overall runtime of this part is

$$\begin{aligned}&\widetilde{O}(\sqrt{n\sum_i p_i}) \cdot (k'\mathcal{T}_\mathsf{K} + k'^2) + k'^2\mathcal{T}_\mathsf{K} + k'^\omega \\ &= \widetilde{O}(n^{0.75}k^{1.25}\epsilon^{-1.25}(\mathcal{T}_\mathsf{K} + k\epsilon^{-1})) + k^2\epsilon^{-2}(\mathcal{T}_\mathsf{K} + k^{\omega-2}\epsilon^{2-\omega}).\end{aligned}$$

- For matrices $C$ and $R$, we do not explicitly compute the data structure for them.
- To form the coreset $\widetilde{R}$ of the columns of $R^\top = D_1^\top K D_2$ using Corollary D.3, we need to show that the generic quantum sampling algorithm could be implemented even though the input is given in factor form. Observe that the algorithm requires uniformly sampling columns of the input matrix, which is oblivious to the input. To form the initial coreset $C_0$, we need to query a total of $\widetilde{O}(\sqrt{nk/\epsilon}) \times \widetilde{O}(k/\epsilon)$ entries of $K$, which can be done in $\widetilde{O}(n^{0.5}k^{1.5}\epsilon^{-1.5})$ kernel evaluations. Then we compute the SVD of this matrix, in time $\widetilde{O}(n^{0.5}k^{\omega-0.5}\epsilon^{0.5-\omega})$. Subsequently we need to implement the classical ridge leverage score data structure (Algorithm 6), which can be done in time $\widetilde{O}(n^{0.5}k^{\omega-0.5}\epsilon^{0.5-\omega})$ and then apply the random Gaussian matrix takes $\widetilde{O}(n^{0.5}k^{1.5}\epsilon^{-1.5})$ time. To implement each query, we can form the query vector by $\widetilde{O}(n^{0.5}k^{0.5}\epsilon^{-0.5})$ kernel evaluations and an additional $\widetilde{O}(n^{0.5}k^{0.5}\epsilon^{-0.5})$ time. The total runtime is

$$\widetilde{O}(n^{0.75}k^{1.25}\epsilon^{-1.25} + n^{0.5}k^{1.5}\epsilon^{-1.5}) \cdot (\mathcal{T}_\mathsf{K} + 1) + \widetilde{O}(n^{0.5}k^{\omega-0.5}\epsilon^{0.5-\omega}).$$

- Form matrix $Z$ by computing SVD of $\widetilde{R}$, since $\widetilde{R} \in \mathbb{R}^{\sqrt{nk/\epsilon}\times k/\epsilon}$, this step could be done in time $O(\epsilon^{0.5-\omega}n^{0.5}k^{\omega-0.5})$.
- Form the sampling matrix $D_3$ involves sampling the $t$ rows of $Z$ according to $p_i = \min\{1, (\epsilon k'/k)\|z_i\|_2^2\}$, where each oracle call returns the squared norm of a stored row of $Z$ in $O(k/\epsilon)$ time, and the sum of $p_i$'s is

$$\sum_{i=1}^{t} p_i \le \frac{\epsilon k'}{k} \cdot \sum_{i=1}^{t} \|z_i\|_2^2 = \frac{\epsilon k'}{k} \cdot \|Z\|_F^2 = k'$$

because $Z$ has $k/\epsilon$ orthonormal columns. Thus, by Lemma B.15 applied to the $t$ row indices of $Z$, the overall runtime of this step is

$$\widetilde{O}(\sqrt{tk'} \cdot k/\epsilon) = \widetilde{O}((nk/\epsilon)^{1/4}(k/\epsilon)^{1/2} \cdot k/\epsilon) = \widetilde{O}(n^{0.25}k^{1.75}\epsilon^{-1.75}),$$

which is dominated by $n^{0.5}k^{1.5}\epsilon^{-2.5}$ since $k \le n$.

- Form matrix $W$, we only need to explicitly compute $(Z^\top D_3)^\dagger \in \mathbb{R}^{k'\times k/\epsilon}$, which is a small matrix and could be computed in time $\widetilde{O}(k^\omega/\epsilon^\omega)$. Note that again, we won't explicitly compute the data structure for $W$, and a row of $W$ is instead formed on demand from $k'$ kernel evaluations.

- Form the leverage score sampling matrix $D_4$ and $D_5$ with respect to $W \in \mathbb{R}^{n \times k/\epsilon}$ and sample $k'/\epsilon^2$ rows/columns. The argument is similar to forming that of $\widetilde{R}$, except we use Algorithm 3, whose approximators are $\widetilde{O}(k/\epsilon) \times k/\epsilon$ matrices formed from rows of $W$. Since we need to oversample $k'/\epsilon^2 = \widetilde{O}(k/\epsilon^3)$ rows and columns, we could scale the scores accordingly and make the sum of probabilities be at most $\widetilde{O}(k/\epsilon^3)$. To implement the oracle, we form the $\widetilde{O}(k/\epsilon)$ rows of each approximator with $\widetilde{O}(k^2\epsilon^{-2})$ kernel evaluations, $k'$ for each row. The subsequent SVDs and applications of the JL matrices take $\widetilde{O}(k^\omega \epsilon^{-\omega})$ time, which is the additive term $d^\omega$ of Theorem C.12 with $d = k/\epsilon$. We multiply the $O(\log n) \times k/\epsilon$ sketching matrices $M$ and $N$ of Algorithm 3 once by $((Z^\top D_3)^\dagger)^\top$, in $\widetilde{O}(k'k/\epsilon)$ time. Since the $i$-th row of $W$ is $w_i = ((Z^\top D_3)^\dagger)^\top v_i$ with $v_i := D_3^\top D_1^\top K_i \in \mathbb{R}^{k'}$, a query on this row is answered by forming $v_i$ with $k'$ kernel evaluations and applying the two precomposed $O(\log n) \times k'$ matrices to it in $\widetilde{O}(k')$ time. Without this precomposition, a query would cost $\widetilde{O}(k'k/\epsilon)$ time. The final Grover search makes $\widetilde{O}(\sqrt{nk'/\epsilon^2})$ queries, and the $O(\log n)$ intermediate levels make fewer. Therefore, the total runtime is

$$\widetilde{O}(\sqrt{nk'/\epsilon^2}) \cdot k'(\mathcal{T}_\mathsf{K} + 1) + \mathrm{poly}(k/\epsilon) = \widetilde{O}(n^{0.5}k^{1.5}\epsilon^{-2.5}) \cdot (\mathcal{T}_\mathsf{K} + 1) + \mathrm{poly}(k/\epsilon).$$

- Form matrix $D_4^\top W$ and $W^\top D_5$ could be done via selecting entries, in time $\widetilde{O}(k^2\epsilon^{-4}) \cdot \mathcal{T}_\mathsf{K}$, and forming the projection matrices $P_{D_4^\top W}$ and $P_{W^\top D_5}$ takes time $\mathrm{poly}(k/\epsilon)$.

- Form the matrix $D_4^\top K D_5$ is selecting its $(k'/\epsilon^2)^2 = \widetilde{O}(k^2\epsilon^{-6})$ entries from $K$, in time $\widetilde{O}(k^2\epsilon^{-6}) \cdot \mathcal{T}_\mathsf{K}$, and this is the largest number of kernel evaluations among the $\mathrm{poly}(k/\epsilon)$ terms.

- Compute $[P_{D_4^\top W}(D_4^\top K D_5)P_{W^\top D_5}]_k$ involves multiplying $\mathrm{poly}(k/\epsilon)$ size matrices and computing an SVD of a $k'/\epsilon^2 \times k'/\epsilon^2$ matrix, which takes $\widetilde{O}((k'/\epsilon^2)^\omega) = \widetilde{O}(k^\omega\epsilon^{-3\omega})$ time, the largest of the $\mathrm{poly}(k/\epsilon)$ arithmetic terms. Forming $Y_*$ and its top-$k$ singular vectors takes $\mathrm{poly}(k/\epsilon)$ time.

- Form the sampling matrix $D_6$ involves performing leverage score sampling according to matrix $WU_* \in \mathbb{R}^{n \times k}$ with the smaller target row count $O(k \log k + k/\epsilon) \leq k'/\epsilon^2$, so the runtime is subsumed by the time to form $D_4$ and $D_5$. Finally, forming the matrix $N$ only requires computing $(D_6^\top W U_*)^\dagger$, which takes $\mathrm{poly}(k/\epsilon) \cdot \mathcal{T}_\mathsf{K}$ time.

Hence, the overall running time of Algorithm 8 is the one stated in the theorem. By the same accounting, the number of kernel evaluations, and hence the number of queries to the points of $X$, is $\widetilde{O}(n^{0.75}k^{1.25}\epsilon^{-1.25} + n^{0.5}k^{1.5}\epsilon^{-2.5} + \mathrm{poly}(k/\epsilon))$. The first of the three terms comes from $D_1$, $D_2$, and $\widetilde{R}$, the second from $D_4$, $D_5$, and $D_6$, and the third from the explicitly formed $\mathrm{poly}(k/\epsilon)$-size submatrices. The $\mathrm{poly}(k/\epsilon)$ term is at least the $\widetilde{O}(k^2\epsilon^{-6})$ entries of $D_4^\top K D_5$. □

# F Quantum $(k, p)$-Subspace Approximation

In this section, we consider a generalized version of the $k$-subspace cost studied in Section C.3, which we call the $(k, p)$-subspace cost [Woodruff and Yasuda, 2025b]: let $\mathcal{F}_k$ be the set of all $k$-dimensional subspaces, then

$$\mathrm{cost}(A, x) = \left(\sum_{i=1}^n \|a_i^\top (I - P_x)\|_2^p\right)^{1/p}.$$

By defining the matrix $(p, 2)$-norm as

$$\|Y\|_{p,2} = \left(\sum_{i=1}^n \|e_i^\top Y\|_2^p\right)^{1/p},$$

we have

$$\mathrm{cost}(A, x) = \|A(I - P_x)\|_{p,2}.$$

The $k$-subspace cost function we studied in Section C.3 is just the $(k, 2)$-subspace cost, and Woodruff and Yasuda [2025b] have shown that, similar to the $k$-subspace cost, one could sample according to the powers of the ridge leverage score. We recall their main result in the following.

**Lemma F.1** (Theorems 3.3 and 3.6 of Woodruff and Yasuda [2025b], one-round form)**.** *Let $A \in \mathbb{R}^{n\times d}$, $\epsilon \in (0,1)$, $\delta \in (0,1)$ and $\alpha = \Theta(\epsilon^2)/(\log^3 n + \log(1/\delta))$. Let $S$ be the $\ell_p$ sampling matrix of Woodruff and Yasuda [2025b] with probabilities $\{q_i\}_{i=1}^n$ (row $i$ is kept independently with probability $q_i$ and rescaled by $q_i^{-1/p}$), where*

$$q_i \geq \begin{cases} \min\{1, n^{p/2-1}\overline{\tau}_i(A, \lambda_{A_k})^{p/2}/\alpha\}, & \text{if } p > 2, \\ \min\{1, \overline{\tau}_i(A, \lambda_{A_{k'}})^{p/2}/\alpha\}, & \text{if } 1 \leq p < 2, \end{cases}$$

*with $\lambda_{A_k} = \|A - A_k\|_F^2/k$ and, for $p < 2$, $\lambda_{A_{k'}} = \|A - A_{k'}\|_F^2/k'$ for a rank parameter $k' \geq k$. Write $L := O(\log(n/(\epsilon\delta)))$ and $\mathrm{OPT} := \min_{x\in\mathcal{F}_k} \|A(I - P_x)\|_{p,2}^p$. Then, with probability at least $1 - \delta$, for every $x \in \mathcal{F}_k$,*

$$\|SA(I - P_x)\|_{p,2}^p = (1 \pm \epsilon)\|A(I - P_x)\|_{p,2}^p \pm O(\epsilon)\, L^{p/2} \cdot \mathrm{OPT} \quad (p > 2),$$
$$\textit{resp.} \quad \|SA(I - P_x)\|_{p,2}^p = (1 \pm \epsilon)\|A(I - P_x)\|_{p,2}^p \pm O(\epsilon)\, L^{p} \cdot \mathrm{OPT} \quad (1 \leq p < 2),$$

*(the statement of Theorem 3.6 in arXiv v3 shows the exponent $p/2$ on its second norm where its proof display and Definition 1.1 have $p$, and we follow the proof) under the following hypotheses:*

- *For $p > 2$ (Theorem 3.3 of Woodruff and Yasuda [2025b]): $n \geq n'$ for some $n' = O(k^{p/2}/\epsilon)$. If $q_i = \min\{1, n^{p/2-1}\widehat{\tau}_i^{p/2}/\alpha\}$ for values $\widehat{\tau}_i \geq \overline{\tau}_i(A, \lambda_{A_k})$, then the expected number of sampled rows is $\sum_i q_i \leq n^{1-2/p}\alpha^{-2/p}\sum_i \widehat{\tau}_i$. For $\widehat{\tau}_i = \overline{\tau}_i(A, \lambda_{A_k})$, this is at most $2kn^{1-2/p}\alpha^{-2/p}$, since these scores sum to at most $2k$ by Lemma C.24. This sharpens the bound $O(kn^{1-2/p}/\alpha)$ in the proof of Theorem 3.5 of Woodruff and Yasuda [2025b], which takes $1/\alpha$ outside the minimum, and we prove it after the lemma.*
- *For $1 \leq p < 2$ (Theorem 3.6 of Woodruff and Yasuda [2025b]): there is a rank-$k'$ subspace $\widetilde{F}$ with $\|a_i^\top(I - P_{\widetilde{F}})\|_2^p \leq O(1/n)\|A(I - P_{\widetilde{F}})\|_{p,2}^p$ for all $i$ and $\|A(I - P_{\widetilde{F}})\|_{p,2}^p \leq O(\mathrm{OPT})$ (a "flat" constant-factor solution). If $q_i = \min\{1, \widehat{\tau}_i^{p/2}/\alpha\}$ for values $\widehat{\tau}_i \geq \overline{\tau}_i(A, \lambda_{A_{k'}})$, then the expected number of sampled rows is $\sum_i q_i \leq n^{1-p/2}(\sum_i \widehat{\tau}_i)^{p/2}/\alpha$ by Hölder's inequality. For $\widehat{\tau}_i = \overline{\tau}_i(A, \lambda_{A_{k'}})$, this is $O(k'^{p/2}n^{1-p/2}/\alpha)$ (proof of Theorem 3.7 of Woodruff and Yasuda [2025b], from $\sum_i \overline{\tau}_i(A, \lambda_{A_{k'}}) \leq 2k'$). Woodruff and Yasuda [2025b] establish the precondition with $k' = \widetilde{O}(k)$ by computing a constant-factor solution and splitting each heavy row into reweighted copies (their Lemmas 2.3 and 2.4).*

*Since $\|A(I - P_x)\|_{p,2}^p \geq \mathrm{OPT}$ for every $x$, replacing $\epsilon$ by $\epsilon/(CL^{p/2})$ (resp. $\epsilon/(CL^p)$) for a suitable constant $C$ turns the guarantee into $\|SA(I-P_x)\|_{p,2}^p = (1\pm\epsilon)\|A(I-P_x)\|_{p,2}^p$ for all $x \in \mathcal{F}_k$, hence $\|SA(I - P_x)\|_{p,2} = (1 \pm \epsilon)\|A(I - P_x)\|_{p,2}$ since $p \geq 1$. This rescaling costs a factor $O(L^p)$ (resp. $O(L^{2p})$) in $1/\alpha$. For $p > 2$, the bound on the number of sampled rows in the first item depends on $\alpha$ only through $\alpha^{-2/p}$, so it grows only by a factor $O(L^2)$. The rescaling also requires, for $p > 2$, the hypothesis $n \geq O(k^{p/2}L^{p/2}/\epsilon)$ in place of $n \geq O(k^{p/2}/\epsilon)$, and the $\widetilde{O}(\cdot)$ below hides these factors. The full procedures of Woodruff and Yasuda [2025b] (Theorems 3.5 and 3.7, $O(\log\log n)$ rounds of this sampling) run in $\widetilde{O}(\mathrm{nnz}(A) + d^\omega)$ time and output coresets of size $k^{p/2}\epsilon^{-p}(\log(n/\delta))^{O(p^2)}$ for $p > 2$ and $k\epsilon^{-4/p}(\log(n/\delta))^{O(1)}$ for $p < 2$.*

To see the bound on $\sum_i q_i$ for $p > 2$ in Lemma F.1, note that $\min\{1, y^{p/2}\} \leq y$ for every $y \geq 0$. Indeed, $y^{p/2} \leq y$ for $y \leq 1$ as $p/2 \geq 1$, and $\min\{1, y^{p/2}\} \leq 1 < y$ for $y > 1$. We apply this with $y = n^{1-2/p}\alpha^{-2/p}\widehat{\tau}_i$, for which $y^{p/2} = n^{p/2-1}\widehat{\tau}_i^{p/2}/\alpha$, and sum over $i$.

Note that the one-round bounds on $\sum_i q_i$ cannot be replaced by the $n$-independent output sizes. To see this, consider $p = 4$ and $k \leq \sqrt{n}$, and let $A \in \mathbb{R}^{n\times d}$ have $N = k\sqrt{n}$ rows forming an

equal-norm tight frame of rank $k$ and $n - N$ zero rows. Then we have $\lambda_{A_k} = 0$, and the zero rows have $q_i = 0$. Each frame row has $\overline{\tau}_i = k/N = 1/\sqrt{n}$ and $n^{p/2-1}\overline{\tau}_i^{p/2} = 1$, so $q_i = 1$ for all $N$ rows and $\sum_i q_i = k\sqrt{n} = kn^{1-2/p}$.

To speed up their algorithm, we note that the dominating runtime part is to sample from the rescaled leverage score distribution, and we could use Theorem C.25 with an inflated sample size.

**Theorem F.2.** *Under the hypotheses of Lemma F.1 ($n \geq \widetilde{O}(k^{p/2}/\epsilon)$ for $p > 2$, none beyond those of Theorem C.25 for $p = 2$, and the flatness precondition at a rank $k' = \widetilde{O}(k)$ for $p \in [1, 2)$) and with $\delta$ a constant, there exists a quantum algorithm that outputs, with probability at least $1 - O(\delta)$, a weighted row sample $SA$ with $\|SA(I - P_x)\|_{p,2}^p = (1 \pm \epsilon)\|A(I - P_x)\|_{p,2}^p$ for all $x \in \mathcal{F}_k$ (hence also $\|SA(I - P_x)\|_{p,2} = (1 \pm \epsilon)\|A(I - P_x)\|_{p,2}$), of expected size $\widetilde{O}(\epsilon^{-4/p}kn^{1-2/p})$ for $p \geq 2$ and $\widetilde{O}(\epsilon^{-2}k^{p/2}n^{1-p/2})$ for $p \in [1, 2)$, and such that*

- *For $p \geq 2$, it runs in expected time $\widetilde{O}(n^{1-1/p}dk^{0.5}\epsilon^{-2/p} + dk^{\omega-1})$ and uses, in expectation, $\widetilde{O}(n^{1-1/p}k^{0.5}\epsilon^{-2/p})$ queries to the points in $A$.*
- *For $p \in [1, 2)$, it runs in expected time $\widetilde{O}(n^{1-p/4}dk^{p/4}\epsilon^{-1} + dk^{\omega-1})$ and uses, in expectation, $\widetilde{O}(n^{1-p/4}k^{p/4}\epsilon^{-1})$ queries to the points in $A$.*

*Here $\widetilde{O}(\cdot)$ hides factors $(\log(n/\epsilon))^{O(p)}$ in the hypothesis $n \geq \widetilde{O}(k^{p/2}/\epsilon)$, and factors $(\log(n/\epsilon))^{O(1)}$ elsewhere.*

*Proof.* We apply Lemma F.1 with $\epsilon$ rescaled as stated there, which makes its guarantee multiplicative, and with $k' = \widetilde{O}(k)$ for $p < 2$, as in the hypothesis. Then, with the probabilities $q_i$ chosen below, the sum of the sampling probabilities is $O(kn^{1-2/p}\alpha^{-2/p})$ for $p > 2$ and $O(k'^{p/2}n^{1-p/2}/\alpha)$ for $p \in [1, 2)$. For constant $\delta$, the rescaled $\alpha$ satisfies $1/\alpha = O(\epsilon^{-2}L^p \log^3 n)$ for $p > 2$, resp. $1/\alpha = O(\epsilon^{-2}L^{2p}\log^3 n) = \widetilde{O}(\epsilon^{-2})$ for $p < 2$. Hence, for $p > 2$, we have $\alpha^{-2/p} = O(\epsilon^{-4/p}L^2(\log n)^{6/p}) = \widetilde{O}(\epsilon^{-4/p})$. In both cases, the hidden factor is $(\log(n/\epsilon))^{O(1)}$. For $p = 2$, the sum is $O(k/\alpha)$ by the definition of $q_i$, and the guarantee is that of Section C.3. Sampling with Lemma B.15 costs $\widetilde{O}(\sqrt{n\sum_i q_i})$ queries with per-query cost $d$, so for $p > 2$ the number of queries is $\widetilde{O}(\sqrt{n \cdot kn^{1-2/p}\epsilon^{-4/p}}) = \widetilde{O}(n^{1-1/p}k^{0.5}\epsilon^{-2/p})$, and for $p \in [1, 2)$ it is $\widetilde{O}(\sqrt{n \cdot k'^{p/2}n^{1-p/2}\epsilon^{-2}}) = \widetilde{O}(n^{1-p/4}k'^{p/4}\epsilon^{-1}) = \widetilde{O}(n^{1-p/4}k^{p/4}\epsilon^{-1})$. The ridge leverage scores at rank $k$, resp. $k'$, are supplied by the oracle of Lemma C.24 (Algorithm 6) built on the constant-accuracy output of Theorem C.25 with rank parameter $\min\{k, d\}$, resp. $\min\{k', d\}$. By Theorem C.25, this output is an approximator, not only a coreset, and for $k' \geq d$ the ridge parameter $\lambda_{A_{k'}}$ vanishes and the scores coincide with those at rank $d$. By Lemma C.21, the QUERY of the oracle returns a constant-factor approximation of $\overline{\tau}_i(A, \lambda_{A_k})$, resp. $\overline{\tau}_i(A, \lambda_{A_{k'}})$, in $\widetilde{O}(d)$ time, and scaling it by a constant gives a value $\widehat{\tau}_i$ between the score and a constant multiple of it. We set $q_i = \min\{1, n^{p/2-1}\widehat{\tau}_i^{p/2}/\alpha\}$, resp. $q_i = \min\{1, \widehat{\tau}_i^{p/2}/\alpha\}$, which is admissible since Lemma F.1 only requires $q_i \geq \min\{1, \cdot\}$. Since $\sum_i \widehat{\tau}_i = O(k)$, resp. $O(k')$, by Lemma C.24, the bounds of Lemma F.1 give the sums of the $q_i$ stated above. Here $k \leq d$, as $\mathcal{F}_k$ consists of $k$-dimensional subspaces of $\mathbb{R}^d$, and we write $k' = k$ for $p \geq 2$. The preprocessing takes $\widetilde{O}(d\min\{k', d\}^{\omega-1})$ time, and so does the additive term of Theorem C.25. Since $k' = \widetilde{O}(k)$, both are $\widetilde{O}(dk^{\omega-1})$. The $\widetilde{O}(n^{0.5}dk'^{0.5})$ cost of Theorem C.25 is dominated by the sampling term, since $1 - 1/p \geq 1/2$, resp. since $n^{0.5}k'^{0.5} \leq n^{1-p/4}k'^{p/4}$ for $k' \leq n$. Up to logarithmic factors, the additive term $dk^{\omega-1}$ is also at most the sampling term whenever $n \geq k^2$. Indeed, the sampling term is at least $n^{0.5}dk^{0.5}$, where for $p < 2$ we use $n^{1/2-p/4} \geq k^{1/2-p/4}$, and for $n \geq k^2$ we have $k^{\omega-1} \leq k^{1.5} \leq n^{0.5}k^{0.5}$ since $\omega < 2.5$. Hence, for $n \geq k^2$, the first term of each running time is its leading term, with no assumption relating $n$ and $d$. Since the algorithm performs one round of sampling, its output has the stated expected size $m$ rather than the $n$-independent size of Woodruff and Yasuda [2025b]. To reach the latter, one can run the algorithm of their Theorems 3.5 and 3.7 on the $m$ weighted sampled rows, where for the $(p, 2)$-norm a weight $w$ rescales a row by $w^{1/p}$. As stated there, this takes $\widetilde{O}(md + d^\omega)$ time. Their proofs do not itemize the running time, and their Section 1.4.2 notes that ridge leverage scores can be approximated up to constant factors in $\widetilde{O}(\mathrm{nnz}(A) + d^\omega)$ time. We

implement every step of their algorithm as follows and obtain $\widetilde{O}(md + dk^{\omega-1})$ time. Their algorithm runs $r = O(\log \log m)$ rounds. For $p < 2$, a round first computes a constant-factor solution $\widetilde{F}$ of the current matrix by their Lemma 2.3 and flattens the matrix with respect to $\widetilde{F}$ by their Lemma 2.4, which multiplies the number of rows by at most $3/2$. The subspace $\widetilde{F}$ is spanned by $\widetilde{O}(k)$ rows, and we take the rank parameter of their Theorem 3.6 to be an upper bound $k'' = \widetilde{O}(k)$ on its dimension, as that theorem requires. We set $k'' = k$ for $p \geq 2$. Then the round applies their Theorem 3.3, resp. 3.6, with probabilities computed from constant-factor approximations of the ridge leverage scores of the current matrix, and keeps a subset of its rows. Hence every matrix that appears has at most $(3/2)^r m = \widetilde{O}(m)$ rows. As above, we approximate the ridge leverage scores of all rows of such a matrix $B$ by writing $B$ into the QRAM in $O(|B|d)$ time, running Theorem C.25 on $B$ at constant accuracy with rank parameter $\min\{k'', d\}$, and querying the oracle of Lemma C.24 built on its output for every row. This takes $\widetilde{O}(|B|^{0.5} dk''^{0.5} + dk^{\omega-1} + |B|d) = \widetilde{O}(|B|d + dk^{\omega-1})$ time. For $p < 2$, their Lemma 2.3 computes $\widetilde{F}$ in $\widetilde{O}(|B|d + k^\omega)$ time, and $k^\omega \leq dk^{\omega-1}$. Their Lemma 2.4 uses the residual norms $\rho_i = \|b_i^\top (I - P_{\widetilde{F}})\|_2$ of the rows $b_i$ of $B$, whose direct computation takes $\widetilde{O}(|B|dk)$ time. Instead, we compute an orthonormal basis $V$ of $\widetilde{F}$ and a sketch $G'(I - VV^\top)$ as in Algorithm 6, with a Gaussian matrix $G'$ of $O(\log(m/\delta))$ rows, in $\widetilde{O}(dk^{\omega-1})$ time. This gives estimates $\widehat{\rho}_i \in [\rho_i/2, 2\rho_i]$ for all rows in $\widetilde{O}(|B|d)$ time, and we run the splitting rule of their Lemma 2.4 with $\widehat{\rho}_i$ in place of $\rho_i$. That is, let $\widehat{R} = \sum_i \widehat{\rho}_i^p$. If $\widehat{R} = 0$, then all $\rho_i$ vanish and $B$ is already flat. Otherwise, each row with $\widehat{\rho}_i^p \geq 2\widehat{R}/|B|$ is replaced by $l_i = \lceil \widehat{\rho}_i^p |B|/(2\widehat{R}) \rceil$ copies of $l_i^{-1/p} b_i$. This preserves $\|B(I - P_F)\|_{p,2}$ for every subspace $F$ and adds at most $\sum_i \widehat{\rho}_i^p |B|/(2\widehat{R}) = |B|/2$ rows. Since $\rho_i^p \leq 2^p \widehat{\rho}_i^p$, each copy satisfies $\|l_i^{-1/p} b_i^\top (I - P_{\widetilde{F}})\|_2^p = \rho_i^p / l_i \leq 2^{p+1} \widehat{R}/|B|$, and each row that is not split satisfies $\rho_i^p < 2^{p+1} \widehat{R}/|B|$. Let $B'$ be the flattened matrix and $R = \sum_i \rho_i^p = \|B'(I - P_{\widetilde{F}})\|_{p,2}^p$. Since $\widehat{R} \leq 2^p R$ and $|B'| \leq 3|B|/2$, every row $b'$ of $B'$ satisfies $\|b'^\top (I - P_{\widetilde{F}})\|_2^p \leq 2^{2p+1} R/|B| \leq 3 \cdot 4^p \|B'(I - P_{\widetilde{F}})\|_{p,2}^p / |B'|$, where $3 \cdot 4^p < 48$. This is the first half of the flatness precondition of their Theorem 3.6. The second half, $\|B'(I - P_{\widetilde{F}})\|_{p,2}^p \leq O(\min_{x \in \mathcal{F}_k} \|B'(I - P_x)\|_{p,2}^p)$, follows from their Lemma 2.3, since flattening preserves all these costs. If each of these subroutines is run with failure probability $\delta/r$ in each round, the reduction fails with probability $O(\delta)$. Summing over the $r$ rounds, it takes $\widetilde{O}(md + dk^{\omega-1})$ time, which is at most the sampling time plus $\widetilde{O}(dk^{\omega-1})$ whenever $n \geq k^{p/2} \epsilon^{-2}$ for $p \geq 2$, resp. $n \geq k\epsilon^{-4/p}$ for $p \in [1, 2)$, since then we have $md \leq n^{1-1/p} dk^{0.5} \epsilon^{-2/p}$, resp. $md \leq n^{1-p/4} dk^{p/4} \epsilon^{-1}$, up to the hidden logarithmic factors. These conditions hold under our assumption that $n \gg \mathrm{poly}(k, 1/\epsilon)$, and composing the two $(1 \pm \epsilon)$ guarantees costs a constant factor in $\epsilon$. Finally, the constant-accuracy approximator of Theorem C.25, the Gaussian sketches of the oracle of Lemma C.24 built on it, the guarantee of Lemma F.1, and the sampler of Lemma B.15 each fail with probability at most $\delta$. For the sketches, the distortion bound of the oracle holds for all $n$ rows since, as in the proof of Theorem C.25, they are taken with $O(\log(n/\delta))$ rows. By a union bound, the success probability is at least $1 - 4\delta = 1 - O(\delta)$, and the running time is an expected time in the sense of the convention following Lemma B.15. For $p \in [1, 2)$, we assume the flatness precondition of Lemma F.1. The step of Woodruff and Yasuda [2025b] that establishes it changes the sampling distribution, and we leave a sublinear-time implementation of it open. □

# G Quantum Tensor Low-Rank Approximation

In this section, we provide a quantum algorithm for computing the Frobenius norm low-rank approximation of a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$. The goal is to find a rank-$k$ tensor $B := \sum_{i=1}^k u_i \otimes v_i \otimes w_i$ for $u_i, v_i, w_i \in \mathbb{R}^n$, such that $\|A - B\|_F^2 \leq (1+\epsilon) \cdot \mathrm{OPT}$ where $\mathrm{OPT} := \inf_{B:\mathrm{rank}(B)=k} \|A - B\|_F^2$. The first caveat is that such an optimal rank-$k$ solution might not even exist. We provide algorithms with constant relative error $(2 + \epsilon)$, specifically a bicriteria algorithm with output rank $k^2$, and a fixed-parameter algorithm with exact output rank $k$ when an optimal rank-$k$ solution exists.

### G.1 Preliminary

Given a 3rd order tensor $A \in \mathbb{R}^{n\times n\times n}$, we define the rank of $A$ as the smallest integer $k$ such that $A = \sum_{i=1}^{k} u_i \otimes v_i \otimes w_i$ where $u_i, v_i, w_i \in \mathbb{R}^n$. We use $\otimes$ to denote the Kronecker product of two matrices, i.e., for $A \in \mathbb{R}^{a\times b}$, $B \in \mathbb{R}^{c\times d}$, $A\otimes B \in \mathbb{R}^{ac\times bd}$ and $A\otimes B = \begin{bmatrix} A_{1,1}B & A_{1,2}B & \dots & A_{1,b}B \\ \vdots & \vdots & \dots & \vdots \\ A_{a,1}B & A_{a,2}B & \dots & A_{a,b}B \end{bmatrix}$. We use $\odot$ to denote a product of two matrices defined as for $A \in \mathbb{R}^{a\times b}$, $B \in \mathbb{R}^{a\times d}$, $A\odot B \in \mathbb{R}^{a\times bd}$ where $A\odot B = \begin{bmatrix} A_{1,*} \otimes B_{1,*} \\ A_{2,*} \otimes B_{2,*} \\ \vdots \\ A_{a,*} \otimes B_{a,*} \end{bmatrix}$, i.e., the matrix formed by computing tensor product between corresponding rows of $A$ and $B$. Given a tensor $A \in \mathbb{R}^{n_1\times n_2\times n_3}$ and three matrices $B_1 \in \mathbb{R}^{n_1\times d_1}$, $B_2 \in \mathbb{R}^{n_2\times d_2}$ and $B_3 \in \mathbb{R}^{n_3\times d_3}$, we define the $(\cdot,\cdot,\cdot)$ operator as

$$A(B_1, B_2, B_3)_{i,j,l} = \sum_{i'=1}^{n_1}\sum_{j'=1}^{n_2}\sum_{l'=1}^{n_3} A_{i',j',l'}(B_1)_{i',i}(B_2)_{j',j}(B_3)_{l',l}, \forall (i,j,l) \in [d_1]\times[d_2]\times[d_3],$$

consequently, $A(B_1, B_2, B_3) \in \mathbb{R}^{d_1\times d_2\times d_3}$. One could also set any of the $B_i$'s as $I_{n_i}$ and for example, $A(B_1, I_{n_2}, I_{n_3}) \in \mathbb{R}^{d_1\times n_2\times n_3}$. When the dimension of the identity matrix is clear from context, we abbreviate it as $I$ for notational simplicity. For $A \in \mathbb{R}^{n_1\times n_2\times n_3}$, we use $A_1 \in \mathbb{R}^{n_1\times n_2 n_3}$, $A_2 \in \mathbb{R}^{n_2\times n_1 n_3}$ and $A_3 \in \mathbb{R}^{n_3\times n_1 n_2}$ to denote the three matrices such that the $[3]\setminus\{i\}$ dimensions are flattened. We use the convention $(A_1)_{i,(j-1)n_3+l} = (A_2)_{j,(i-1)n_3+l} = (A_3)_{l,(i-1)n_2+j} = A_{i,j,l}$. Under this convention, for the design matrices $B_m$ in Algorithm 12 and $Z_j$ in the proof of Theorem G.7 below, the $i$-th row is the Kronecker product of the $i$-th columns of the two remaining factors in increasing mode order. For factor matrices $U, V, W \in \mathbb{R}^{n\times k}$ with columns $U_i, V_i, W_i$, we write $U\otimes V\otimes W := \sum_{i=1}^{k} U_i \otimes V_i \otimes W_i \in \mathbb{R}^{n\times n\times n}$ for the tensor with these factors, as in Song et al. [2019]. Note that this is not the Kronecker product of the three matrices.

We also state an algorithm due to Song et al. [2019] for sampling according to leverage scores of $U \odot V$:

**Lemma G.1.** *Given two matrices $U \in \mathbb{R}^{k\times n_1}$ and $V \in \mathbb{R}^{k\times n_2}$, there exists an algorithm*

$$\textsc{TensorLeverageScore}(U, V, n_1, n_2, k, \epsilon, R_{\text{sample}})$$

*that takes*

$$O((n_1+n_2)\cdot \text{poly}(\log(n_1 n_2), k, \epsilon^{-1})\cdot R_{\text{sample}})$$

*time to generate a weighted sampling matrix $D \in \mathbb{R}^{n_1 n_2\times R_{\text{sample}}}$ according to a constant-factor approximation of the leverage score distribution of the columns of $U \odot V$, with probability at least $1 - 1/\text{poly}(n_1 n_2)$ minus the failure probability of its internal sketch, which can be made any constant at a constant factor in the* poly *term. (This is Lemma C.32 and Algorithm 10 of Song et al. [2019]. When $U \odot V$ is rank-deficient, the matrix $R^{-1}$ used there is read as the pseudo-inverse of a rank-revealing factorization, which samples by the leverage scores of the row space of $U \odot V$.)*

To obtain our fixed-parameter tractable algorithm for rank-$k$ tensor low-rank approximation, we require the following result from Song et al. [2019]:

**Lemma G.2.** *Let $\max\{t_i, d_i\} \le n$, given a $t_1 \times t_2 \times t_3$ tensor $A$ and three matrices: $T_1 \in \mathbb{R}^{t_1\times d_1}, T_2 \in \mathbb{R}^{t_2\times d_2}$ and $T_3 \in \mathbb{R}^{t_3\times d_3}$, if for any $\delta > 0$ there exists a solution to*

$$\min_{X_1,X_2,X_3} \| \sum_{i=1}^{k} (T_1X_1)_i \otimes (T_2X_2)_i \otimes (T_3X_3)_i - A\|_F^2 := \text{OPT},$$

*and each entry of $X_i$ could be expressed with $O(n^\delta)$ bits (and each entry of $A$ and of $T_1, T_2, T_3$ is a rational number of $O(n^\delta)$ bits), then there exists an algorithm that takes $n^{O(\delta)}\cdot 2^{O(d_1k+d_2k+d_3k)}$ time and outputs three matrices $\widehat{X}_1, \widehat{X}_2$ and $\widehat{X}_3$ such that $\| \sum_{i=1}^{k} (T_1\widehat{X}_1)_i \otimes (T_2\widehat{X}_2)_i \otimes (T_3\widehat{X}_3)_i - A\|_F^2 =$ OPT.*

**Remark G.3.** *We read the proof of Theorem C.45 of Song et al. [2019] as follows. It binary-searches the cost, with range and precision fixed by the bit-size lower bound of their Theorem B.11, and at each step it decides whether the polynomial system "cost $\leq$ guess" has a solution in the ball $\|x\|_2^2 \leq 2^{O(n^\delta)}$. Therefore, its output has cost at most that of* any *point of that ball with entries of $O(n^\delta)$ bits, up to an additive $2^{-\Omega(n^\delta)}$. The lemma as printed states the case in which such a point is a minimizer, and we use it in the general form, as Lemma G.6 does.*

### G.2 A $(2+\epsilon)$ Bicriteria Algorithm with Output Rank $k^2$

In the first stage, we fix a $k$-dimensional span $Q$ for the first factors. We then treat the second and third modes *jointly* as one matrix low-rank approximation problem on the $n \times kn$ matrix obtained by contracting $A$ with $Q$, and we solve it using the column coreset of Corollary D.3. A rank-$k$ factorization of this matrix unfolds to a tensor of CP rank $k^2$, with only $k$ distinct first factors.

**Algorithm 10** Quantum $(2+\epsilon)$ bicriteria tensor low-rank approximation with output rank $k^2$.

**procedure** QBICRITERIATENSORLOWRANKTWO($A \in \mathbb{R}^{n\times n\times n}, k, \epsilon$)
  $C_1 \leftarrow$ QLOWRANKCMM($A_1, k, \epsilon/3, 0.001$) ▷ $C_1 \in \mathbb{R}^{n\times s_1}$, $s_1 = \widetilde{O}(k/\epsilon^2)$, by Corollary D.3.
  $Q \leftarrow$ top-$k$ left singular vectors of $C_1$ ▷ $Q \in \mathbb{R}^{n\times k}$, $Q^\top Q = I_k$.
  $S_0 \leftarrow$ leverage score sampling matrix of the rows of $Q$ with $m = O(k\log k + k/\epsilon^2)$ rows ▷ Lemma B.13.
  $\widetilde{Q}^\top \leftarrow (S_0 Q)^\dagger S_0$ ▷ $\widetilde{Q}^\top \in \mathbb{R}^{k\times n}$.
  Let $\widetilde{M} \in \mathbb{R}^{n\times kn}$ be the mode-2 unfolding of $A(\widetilde{Q}, I, I)$, i.e., $\widetilde{M}_{j,(t,l)} = \sum_{i\in \mathrm{supp}(S_0)} \widetilde{Q}_{i,t} A_{i,j,l}$ ▷ accessed implicitly
  $C_M \leftarrow$ QLOWRANKCMM($\widetilde{M}, k, \epsilon/3, 0.001$) ▷ $C_M \in \mathbb{R}^{n\times s}$, $s = \widetilde{O}(k/\epsilon^2)$ columns of $\widetilde{M}$.
  $V \leftarrow$ top-$k$ left singular vectors of $C_M$ ▷ $V \in \mathbb{R}^{n\times k}$.
  $S_1 \leftarrow$ leverage score sampling matrix of the rows of $V$ with $m_1 = O(k\log k + k/\epsilon)$ rows
  $Y \leftarrow (S_1 V)^\dagger S_1 \widetilde{M}$ ▷ $Y \in \mathbb{R}^{k\times kn}$.
  For $i, t \in [k]$ let $y_{i,t} \in \mathbb{R}^n$ be the vector $(Y_{i,(t,l)})_{l\in[n]}$
  **return** $Q, V, \{y_{i,t}\}_{i,t\in[k]}$, representing $B = \sum_{i,t\in[k]} Q_t \otimes V_i \otimes y_{i,t}$
**end procedure**

**Theorem G.4.** *Given a 3rd order tensor $A \in \mathbb{R}^{n\times n\times n}$, a positive integer $k \leq n$ and $\epsilon \in (0, 0.1)$, let $\mathrm{OPT} := \inf_{\text{rank-}k\ A_k} \|A - A_k\|_F^2$. Algorithm 10 outputs $Q, V \in \mathbb{R}^{n\times k}$ and vectors $y_{i,t} \in \mathbb{R}^n$, $i, t \in [k]$, such that the tensor $B = \sum_{i,t\in[k]} Q_t \otimes V_i \otimes y_{i,t}$, of CP rank at most $k^2$ with only $k$ distinct first factors, satisfies*

$$\|A - B\|_F^2 \leq (2+\epsilon)\cdot \mathrm{OPT}$$

*with probability at least* 0.9*. It runs in time*

$$\widetilde{O}(\epsilon^{-1} n^2 k^{0.5} + \epsilon^{-3} n^{1.5} k^2 + \epsilon^{-3} n k^3 + \epsilon^{2-2\omega} n k^{\omega-1}) = \widetilde{O}(\epsilon^{-1} n^2 k^{0.5} + n^{1.5}\,\mathrm{poly}(k/\epsilon)),$$

*using $\widetilde{O}(\epsilon^{-1} n k^{0.5})$ queries to mode-1 fibers of $A$ (the columns of $A_1$) and $\widetilde{O}(\epsilon^{-3} n^{0.5} k^2)$ queries to mode-2 and mode-3 fibers.*

*Proof.* We run Algorithm 10 with accuracy parameter $\epsilon_0 := \epsilon/13$ and prove $\|A-B\|_F^2 \leq (2+13\epsilon_0)\cdot \mathrm{OPT}$. This rescaling and the accuracy $\epsilon/3$ of the two coreset calls change the running time and query bounds only by constant factors, which are absorbed by the $\widetilde{O}(\cdot)$. We write $\epsilon$ for $\epsilon_0$ throughout the proof, and fix $\eta > 0$ and a rank-$k$ tensor $B^\star = \sum_{i=1}^k u_i \otimes v_i \otimes w_i$ with $\|A - B^\star\|_F^2 \leq (1+\epsilon)\,\mathrm{OPT}'$, where $\mathrm{OPT}' := \mathrm{OPT} + \eta$. Such a $B^\star$ exists by the definition of OPT as an infimum, even when $\mathrm{OPT} = 0$ is not attained, and if the infimum is attained, one may take $\eta = 0$. Since the output $B$ of the algorithm does not depend on $B^\star$, the bound $\|A - B\|_F^2 \leq (2+13\epsilon)(\mathrm{OPT} + \eta)$ for every $\eta > 0$ gives the claim, and hence below we write OPT for $\mathrm{OPT}'$. The unfolding $B_1^\star$ of $B^\star$ has matrix rank at most $k$, so we have $\|A_1 - [A_1]_k\|_F^2 \leq \|A_1 - B_1^\star\|_F^2 = \|A - B^\star\|_F^2 \leq (1+\epsilon)\,\mathrm{OPT}$.

*Step 1: the first-factor span.* By Corollary D.3, $C_1 \subseteq A_1$ is an $(\epsilon/3)$-coreset of the columns of $A_1$ for the $k$-subspace cost. Therefore, by Lemma 3 of Cohen et al. [2015a], as stated in Section D, the projection $QQ^\top$ onto the top-$k$ left singular vectors of $C_1$ satisfies

$$e_1 := \|A_1 - QQ^\top A_1\|_F^2 \le (1+\epsilon)\|A_1 - [A_1]_k\|_F^2 \le (1+\epsilon)^2 \,\mathrm{OPT}\,.$$

*Step 2: Pythagoras.* Since $Q$ has orthonormal columns, $(I - QQ^\top)A_1$ is orthogonal to $QY$ for every $Y \in \mathbb{R}^{k\times n^2}$, and hence we have

$$\|A_1 - QY\|_F^2 = e_1 + \|Q^\top A_1 - Y\|_F^2 \qquad \text{for all } Y \in \mathbb{R}^{k\times n^2}. \tag{15}$$

*Step 3: reshaping.* The matrix $Q^\top A_1 \in \mathbb{R}^{k\times n^2}$ is the mode-1 unfolding of the $k \times n \times n$ tensor $A(Q, I, I)$, and we denote the mode-2 unfolding of this tensor by $M \in \mathbb{R}^{n\times kn}$, i.e., $M_{j,(t,l)} = \sum_i Q_{i,t}A_{i,j,l}$. Unfoldings preserve Frobenius norms. The tensor $B^\star(Q, I, I) = \sum_{i=1}^k (Q^\top u_i) \otimes v_i \otimes w_i$ has mode-2 unfolding $M^\star = \sum_{i=1}^k v_i\big((Q^\top u_i) \otimes w_i\big)^\top$, whose rank is at most $k$, and we have

$$\|M - M^\star\|_F^2 = \|Q^\top(A_1 - B_1^\star)\|_F^2 \le \|A_1 - B_1^\star\|_F^2 \le (1+\epsilon)\,\mathrm{OPT},$$

since $Q^\top$ is a contraction. Conversely, let $V \in \mathbb{R}^{n\times k}$ and $Y \in \mathbb{R}^{k\times kn}$, and for $i \in [k]$, let $Y^{(i)} \in \mathbb{R}^{k\times n}$ be the $i$-th row of $Y$ reshaped with $Y^{(i)}_{t,l} = Y_{i,(t,l)}$. Then $VY$ is the mode-2 unfolding of the $k \times n \times n$ tensor $\sum_{i,t\in[k]} e_t \otimes V_i \otimes (Y^{(i)}_{t,*})^\top$, and applying $Q$ to the first mode gives $B = \sum_{i,t\in[k]} Q_t \otimes V_i \otimes y_{i,t}$ with $y_{i,t} = (Y^{(i)}_{t,*})^\top$. The mode-1 unfolding of $B$ is $Q \cdot Y'$, where $Y'$ is the mode-1 unfolding of the $k \times n \times n$ tensor above. Therefore, by Eq. (15), we have

$$\|A - B\|_F^2 = e_1 + \|M - VY\|_F^2. \tag{16}$$

Moreover, $B$ has CP rank at most $k^2$ and only $k$ distinct first factors.

*Step 4: implicit access.* A column $(t, l)$ of $M$ depends on the whole frontal slice $A_{*,*,l}$, which is too expensive to read. Instead, $S_0$ samples $m = O(k\log k + k/\epsilon^2)$ rows of $Q$ according to the leverage scores of $Q$, and we apply Lemma B.13 with accuracy parameter $\epsilon^2$, design $Q$ of rank $k$, and response $A_1$. Thus, $Y_0 := (S_0Q)^\dagger S_0A_1 = \widetilde{Q}^\top A_1$ satisfies $\|QY_0 - A_1\|_F^2 \le (1+\epsilon^2)\min_Y \|QY - A_1\|_F^2 = (1+\epsilon^2)e_1$, and by Eq. (15) with $Y = Y_0$, we have $\|Q^\top A_1 - \widetilde{Q}^\top A_1\|_F^2 \le \epsilon^2 e_1$. Note that the matrix $\widetilde{M}$ of the algorithm is the mode-2 unfolding of $A(\widetilde{Q}, I, I)$, hence

$$\|\widetilde{M} - M\|_F^2 = \|(\widetilde{Q}^\top - Q^\top)A_1\|_F^2 \le \epsilon^2 e_1 \le \epsilon^2(1+\epsilon)^2\,\mathrm{OPT}\,.$$

Since $\widetilde{Q}^\top$ is supported on the $m$ rows in $\mathrm{supp}(S_0)$, an entry $\widetilde{M}_{j,(t,l)}$ is computed from the $m$ entries $A_{i,j,l}$, $i \in \mathrm{supp}(S_0)$. Hence, a column $(t, l)$ of $\widetilde{M}$ is computed from the $m$ mode-2 fibers $A_{i,*,l}$, a row $j$ from the $m$ mode-3 fibers $A_{i,j,*}$, and all three fiber types are admissible queries, see Section K. We write the $mk$ entries of $\widetilde{Q}^\top$ to QRAM once, and then a column of $\widetilde{M}$ can be formed inside the sampler of Lemma B.15. Every bound below is routed through $M$ and never through $\widetilde{Q}^\top(A_1 - B_1^\star)$, because $\widetilde{Q}^\top$ is not a contraction.

*Step 5: the matrix problem.* Corollary D.3 applied to $\widetilde{M}$, with $d = n$ rows and $kn$ columns, returns an $(\epsilon/3)$-coreset $C_M$ of the columns of $\widetilde{M}$ for the $k$-subspace cost. Let $V$ be the top-$k$ left singular vectors of $C_M$. By Lemma 3 of Cohen et al. [2015a], we have

$$\|\widetilde{M} - VV^\top\widetilde{M}\|_F^2 \le (1+\epsilon)\|\widetilde{M} - \widetilde{M}_k\|_F^2 \le (1+\epsilon)\|\widetilde{M} - M^\star\|_F^2,$$

using $\mathrm{rank}(M^\star) \le k$. Next, $S_1$ samples $m_1 = O(k\log k + k/\epsilon)$ rows of $V$ according to its leverage scores, and by Lemma B.13 with design $V$ and response $\widetilde{M}$, the matrix $Y = (S_1V)^\dagger S_1\widetilde{M}$ satisfies

$$\|VY - \widetilde{M}\|_F^2 \le (1+\epsilon)\min_{Y'}\|VY' - \widetilde{M}\|_F^2 = (1+\epsilon)\|\widetilde{M} - VV^\top\widetilde{M}\|_F^2 \le (1+\epsilon)^2\|\widetilde{M} - M^\star\|_F^2.$$

By the triangle inequality and the bounds of Steps 3 and 4, for $\epsilon \le 0.1$ we have

$$\|\widetilde{M} - M^\star\|_F \le \|\widetilde{M} - M\|_F + \|M - M^\star\|_F$$

$$\leq \big(\epsilon(1+\epsilon) + \sqrt{1+\epsilon}\big)\sqrt{\mathrm{OPT}} \leq (1+1.6\epsilon)\sqrt{\mathrm{OPT}},$$

so $\|\widetilde{M} - M^\star\|_F^2 \leq (1+3.5\epsilon)\,\mathrm{OPT}$, and consequently $\|VY - \widetilde{M}\|_F^2 \leq (1+\epsilon)^2(1+3.5\epsilon)\,\mathrm{OPT} \leq (1+6.4\epsilon)\,\mathrm{OPT}$. Then

$$\begin{aligned}\|M - VY\|_F &\leq \|M - \widetilde{M}\|_F + \|\widetilde{M} - VY\|_F \\ &\leq \big(\epsilon(1+\epsilon) + \sqrt{1+6.4\epsilon}\big)\sqrt{\mathrm{OPT}} \leq (1+4.3\epsilon)\sqrt{\mathrm{OPT}},\end{aligned}$$

hence $\|M - VY\|_F^2 \leq (1+10.5\epsilon)\,\mathrm{OPT}$ for $\epsilon \leq 0.1$. By Eq. (16) and Step 1, we have

$$\|A - B\|_F^2 \leq (1+\epsilon)^2\,\mathrm{OPT} + (1+10.5\epsilon)\,\mathrm{OPT} \leq (2+13\epsilon)\,\mathrm{OPT}\,.$$

The failure events are the two calls of Corollary D.3, each with probability $0.001$, and the two applications of Lemma B.13, each run with failure probability at most $0.04$ by Remark B.14, so a union bound gives success probability at least $0.9$.

*Running time and queries.* By Corollary D.3 with $d = n$ and $n^2$ columns, computing $C_1$ takes $\widetilde{O}(\epsilon^{-1}n^2k^{0.5} + nk^{\omega-1})$ time and $\widetilde{O}(\epsilon^{-1}nk^{0.5})$ queries to the columns of $A_1$, i.e., mode-1 fibers of $A$. The SVD of $C_1 \in \mathbb{R}^{n\times s_1}$ takes $\widetilde{O}(ns_1^{\omega-1}) = \widetilde{O}(\epsilon^{2-2\omega}nk^{\omega-1})$ time. The leverage scores of $Q$ are its squared row norms, which are computed in $O(nk)$ time, and $(S_0Q)^\dagger$ costs $\mathrm{poly}(k/\epsilon)$. By Theorem C.3 with the oracle of Lemma C.24, the call $\textsc{QLowRankCMM}(\widetilde{M}, k, \epsilon/3, 0.001)$ accesses $\widetilde{O}(\sqrt{kn\cdot s}) = \widetilde{O}(\epsilon^{-1}n^{0.5}k)$ columns of $\widetilde{M}$, where $s = \widetilde{O}(k/\epsilon^2)$. Each column is formed from $m$ mode-2 fibers in $O(nm) = \widetilde{O}(\epsilon^{-2}nk)$ time, and this gives a total of $\widetilde{O}(\epsilon^{-3}n^{1.5}k^2)$ time and $\widetilde{O}(\epsilon^{-3}n^{0.5}k^2)$ mode-2 fiber queries. The oracle preprocessing of this call takes $\widetilde{O}(nk^{\omega-1})$ time per level for the SVDs of the $n \times \widetilde{O}(k)$ approximators, each oracle query on an already formed column takes $\widetilde{O}(n)$ time, which is dominated by forming the column, and the SVD of $C_M \in \mathbb{R}^{n\times s}$ takes $\widetilde{O}(\epsilon^{2-2\omega}nk^{\omega-1})$ time. For these SVDs, we materialize the $\widetilde{O}(k)$ approximator columns per level and the $s$ coreset columns of $\widetilde{M}$, which costs $\widetilde{O}(s\cdot nm) = \widetilde{O}(\epsilon^{-4}nk^2)$ time and $\widetilde{O}(sm)$ mode-2 fiber queries. This is at most the sampling cost above when $n \geq \epsilon^{-2}$. When $n < \epsilon^{-2}$, at most $\min\{s, kn\}$ columns are materialized, which takes $\widetilde{O}(\epsilon^{-2}n^2k^2) \leq \widetilde{O}(\epsilon^{-3}n^{1.5}k^2)$ time and $\widetilde{O}(\epsilon^{-2}nk^2)$ queries. Computing $Y$ requires the $m_1$ sampled rows of $\widetilde{M}$, and each of them is formed from $m$ mode-3 fibers in $O(nkm)$ time. This takes $\widetilde{O}(m_1\cdot nkm) = \widetilde{O}(\epsilon^{-3}nk^3)$ time and $\widetilde{O}(\epsilon^{-3}k^2)$ mode-3 fiber queries, plus $\widetilde{O}(m_1k\cdot kn)$ time for the product $(S_1V)^\dagger S_1\widetilde{M}$. Summing up gives the stated time, and the queries are $\widetilde{O}(\epsilon^{-1}nk^{0.5})$ mode-1 fibers and $\widetilde{O}(\epsilon^{-3}n^{0.5}k^2)$ mode-2 and mode-3 fibers. □

**Remark G.5** (A limitation of this route)**.** *The route of Theorem G.4 cannot reach $(1+\epsilon)$, since its first stage fixes the span of the first factors to a $k$-dimensional subspace $\mathrm{span}(Q)$. Consider $k=1$, $n = m+1$, $A_{1,1,1} = 1$, and $A_{2,j,j} = c/\sqrt{m}$ for $j = 2,\dots,n$, where $1 < c \leq \sqrt{m}$ and all other entries are zero. Then $\mathrm{OPT} = c^2$, attained by $e_1\otimes e_1\otimes e_1$, and approximating the diagonal slice instead costs at least $1 + c^2(1-1/m) \geq c^2$. Once $c^2(1-\epsilon) > 1+\epsilon$, the first stage returns $Q = e_2$ with residual $\|A_1 - QQ^\top A_1\|_F^2 = 1$, and $Q^\top A_1$ reshapes to $(c/\sqrt{m})\cdot I_m$ padded with a zero row and column, whose best rank-$r$ approximation has error $c^2(1-r/m)$. Therefore, every output tensor of CP rank $r$ with first factors in $\mathrm{span}(Q)$ costs at least $1 + c^2(1-r/m)$, which is a ratio of at least $1/c^2 + 1 - r/m$ to $\mathrm{OPT}$. This ratio tends to $1 + 1/c^2$ as $m\to\infty$, and if in addition $c^2 \downarrow (1+\epsilon)/(1-\epsilon)$, it tends to $2 - 2\epsilon/(1+\epsilon)$, which exceeds $1+\epsilon$ for every $\epsilon \in (0, 0.1]$. This is a limitation of the route and not of the problem, and whether $(1+\epsilon)$ is reachable in sublinear time remains open, see Section 3.*

### G.3 Quantum Tensor Low-Rank Approximation: Fixed-Parameter Tractable Algorithm

Reusing the first stage of Section G.2, we fix a $k$-dimensional span $Q$ for the first factors and form the $k\times n\times n$ tensor $A(\widetilde{Q}, I, I)$ explicitly. We then solve the remaining rank-$k$ problem to $(1+\epsilon)$ using the classical fixed-parameter algorithm of Song et al. [2019] cited in Table 2. The output has exact rank $k$ and ratio $(2+\epsilon)$. We import the classical algorithm in the following form.

**Lemma G.6** (Competitor-relative form of the proof method of Theorem C.2 (I) in Song et al. [2019])**.** *Let $A' \in \mathbb{R}^{n\times n\times n}$ have entries of $O(n^\delta)$ bits (fixed-point, i.e., integers over a common power-of-two denominator, which is how we read the rounding of Song et al. [2019] to "an integer*

*expressed using $O(n^{\delta_2'})$ bits"), let $k \geq 1$, $\epsilon \in (0,1)$ and $n^{\delta_2'} = C_0\, n^{\delta} 2^{O(k^2/\epsilon)}$ for a sufficiently large constant $C_0$ (the parameter of Song et al. [2019]), and let $B' = \sum_{i=1}^{k} u_i' \otimes v_i' \otimes w_i'$ be any fixed rank-$k$ tensor whose factor matrices $U', V', W' \in \mathbb{R}^{n\times k}$ satisfy $\max\{\|U'\|_F, \|V'\|_F, \|W'\|_F\} \leq 2^{O(n^{\delta_2'})}$ ($B'$ is not given to the algorithm). There is a classical randomized algorithm that runs in $(\operatorname{nnz}(A') + n\,\mathrm{poly}(k/\epsilon) + 2^{O(k^2/\epsilon)}) \cdot n^{O(\delta)}$ time in the unit cost RAM model with $O(\log n)$-bit words and outputs $U, V, W \in \mathbb{R}^{n\times k}$ with entries of $n^{\delta_2'}$ bits such that $\|U \otimes V \otimes W - A'\|_F^2 \leq (1+\epsilon)\|B' - A'\|_F^2 + 2^{-\Omega(n^{\delta_2'})} \max\{1, \|A'\|_F^2\}$ with probability at least $9/10$.*

Theorem C.2 of Song et al. [2019] is stated relative to $\mathrm{OPT}(A') = \inf_{\text{rank-}k} \|A_k - A'\|_F^2$ under attainment hypotheses, and Theorem C.1 there is stated relative to the minimum over rank-$k$ tensors. We import the proof method of Part (I) of Theorem C.2, in which the factors of a fixed rank-$k$ tensor are rounded to $O(n^{\delta_2'})$-bit integers. The rounding gives the reduced problem a near-optimal solution of bounded bit size, which is the hypothesis of the polynomial system verifier of Lemma G.2 (Theorem C.45 of Song et al. [2019]), read as in Remark G.3. The sketched regressions of Theorem C.1 are then run with those fixed factors, and they yield a rank-$k$ output within $(1+\epsilon)$ of the cost of the rounded tensor. In Song et al. [2019], the fixed tensor is an optimal $A_k$, and the rounding loss is absorbed through a lower bound on $\mathrm{OPT}(A')$. We instead fix an arbitrary bounded-norm competitor and keep the rounding loss as the additive term, as Song et al. [2019] themselves do in their Theorem 1.2 and in Part (IV) of Theorem C.2, where, as they explain, an additive $2^{-\mathrm{poly}(n)}$ is needed to write down the solution in the unit cost RAM model, since in general its entries may be irrational. We use the lemma with the competitor exhibited in the proof below and absorb the additive term there. Note that the rank-$k$ infimum of the tensor we hand to the algorithm need not be attained.

**Algorithm 11** Quantum $(2+\epsilon)$ FPT rank-$k$ low-rank approximation.

1: **procedure** QFPTLowRankTwo($A, k, \epsilon$) ▷ Theorem G.7
2: $C_1 \leftarrow$ QLowRankCMM($A_1, k, \epsilon/3, 0.001$) and $Q \leftarrow$ top-$k$ left singular vectors of $C_1$ ▷ As in Algorithm 10.
3: $Q' \leftarrow Q$ with entries rounded to $cn^{\delta_2'}$ bits after the binary point, $S_0 \leftarrow$ leverage score sampling matrix of the rows of $Q'$ with $m = O(\epsilon^{-2} k \log k)$ expected rows and sampling probabilities floored at $n^{-2}$, and $\widetilde{Q}^{\top} \leftarrow (S_0 Q')^{\dagger} S_0$, entries rounded to $c\,n^{\delta} + \lceil \log_2(100/\epsilon) \rceil$ bits after the binary point ($c \geq 2$)
4: Form $\widetilde{A} := A(\widetilde{Q}, I, I) \in \mathbb{R}^{k\times n\times n}$ explicitly: $\widetilde{A}_{t,j,l} = \sum_{i\in \mathrm{supp}(S_0)} \widetilde{Q}_{i,t} A_{i,j,l}$ ▷ Reads the $m$ slabs $A_{i,*,*}$, $i \in \mathrm{supp}(S_0)$.
5: **for** $j = 1, 2, 3$ **do**
6: $(X_j, V_j, W_j) \leftarrow$ the classical algorithm of Lemma G.6 on $\widetilde{A}$ (whose entries are fixed-point numbers of $O(n^{\delta})$ bits, formed from the rounded $\widetilde{Q}^{\top}$, and which is padded by zeros to $n \times n \times n$), with $X_j \in \mathbb{R}^{k\times k}$ the first $k$ rows of the returned $n \times k$ first factor ▷ $V_j, W_j \in \mathbb{R}^{n\times k}$
7: $\widehat{c}_j \leftarrow \|G(\widetilde{A}_{(2)} - (T_j)_{(2)})\|_F^2$ for a Rademacher matrix $G \in \mathbb{R}^{s\times n}$ with i.i.d. $\pm 1$ entries scaled by $s^{-1/2}$, $s \geq 48000\,\epsilon^{-2}$ ▷ $T_j := \sum_{i=1}^{k} (X_j)_i \otimes (V_j)_i \otimes (W_j)_i$
8: **end for**
9: $j^{\star} \leftarrow \arg\min_j \widehat{c}_j$
10: **return** $U := Q' X_{j^{\star}} \in \mathbb{R}^{n\times k}$, $V_{j^{\star}}, W_{j^{\star}}$
11: **end procedure**

*Note on Algorithm 11.* In Line 7, $(\cdot)_{(2)}$ denotes the mode-2 unfolding, which is an $n \times kn$ matrix.

**Theorem G.7.** *Let $k \geq 1$ be an integer, let $A \in \mathbb{R}^{n\times n\times n}$ have entries of $O(n^{\delta})$ bits for some $\delta > 0$, where we assume $n^{\delta} \geq \log n$ as in Song et al. [2019], let $\mathrm{OPT} := \inf_{\text{rank-}k\ A_k} \|A - A_k\|_F^2 > 0$ be attained by $A_k = U^{\star} \otimes V^{\star} \otimes W^{\star}$ with $\max\{\|U^{\star}\|_F, \|V^{\star}\|_F, \|W^{\star}\|_F\} \leq 2^{O(n^{\delta_2'})}$, where $n^{\delta_2'} = C_0\, n^{\delta} 2^{O(k^2/\epsilon)}$ for a sufficiently large constant $C_0$ is the parameter of Lemma G.6, and assume $\mathrm{OPT} \geq 2^{-n^{\delta}} \|A\|_F^2$. For $\epsilon \in (0, 0.1)$ with $\epsilon \geq 2^{-n^{\delta}}$, Algorithm 11 outputs $n \times k$ matrices $U, V, W$ with*

$$\|U \otimes V \otimes W - A\|_F^2 \leq (2+\epsilon) \cdot \mathrm{OPT}$$

*with probability at least* $0.99$*, in* $(n^2k^2/\epsilon^2 + n\,\mathrm{poly}(k/\epsilon) + 2^{O(k^2/\epsilon)}) \cdot n^{O(\delta)}$ *time in the unit-cost RAM model with* $O(\log n)$*-bit words of Lemma G.6 (the word-level reading of the QRAM model of the introduction, and bit-level counts differ by an* $O(\log n)$ *factor absorbed in* $n^{O(\delta)}$*), using* $\widetilde{O}(\epsilon^{-1}nk^{0.5})$ *queries to mode-*$1$ *fibers and* $\widetilde{O}(\epsilon^{-2}nk)$ *queries to mode-*$3$ *fibers of* $A$*. Without the hypothesis* $\mathrm{OPT} \geq 2^{-n^\delta}\|A\|_F^2$*, carrying the rounding perturbation* $\eta = 2^{-\Omega(n^\delta)}\|A\|_F$ *and the additive term of Lemma G.6 through the two triangle inequalities of the proof and using* $2ab \leq \epsilon a^2 + b^2/\epsilon$ *gives, after a further constant rescaling of* $\epsilon$*,* $\|U \otimes V \otimes W - A\|_F^2 \leq (2+\epsilon)\,\mathrm{OPT} + \epsilon^{-1}2^{-\Omega(n^\delta)}\|A\|_F^2 + 2^{-\Omega(n^{\delta_2'})}\max\{1, \|A\|_F^2\}$ *(the* $\epsilon^{-1}$ *on the last term and the polynomial factor relating* $\|\widetilde{A}'\|_F$ *to* $\|A\|_F$ *are absorbed using* $\epsilon \geq 2^{-n^\delta}$*). The cases in which no optimal rank-*$k$ *solution exists or* $\mathrm{OPT} = 0$ *are not treated.*

*Proof.* We run the algorithm with accuracy parameter $\epsilon_0 := \epsilon/14$ and prove the bound $(2 + 14\epsilon_0)\,\mathrm{OPT}$. The rescaling changes the running time only by constants, and the hypotheses $\epsilon \geq 2^{-n^\delta}$ and $n^{\delta_2'} = C_0\, n^\delta 2^{O(k^2/\epsilon)}$ transfer to $\epsilon_0$ up to the constant $14$, which is absorbed by the $O(\cdot)$'s. Throughout the proof, we write $\epsilon$ for $\epsilon_0$ and let $B^\star := A_k$.

*Steps 1–4 of the proof of Theorem G.4* apply verbatim. By Step 1, we have $e_1 := \|A_1 - QQ^\top A_1\|_F^2 \leq (1+\epsilon)^2\,\mathrm{OPT}$. Let $\widetilde{A} = A(\widetilde{Q}, I, I)$ and $A(Q, I, I)$ be the tensors whose mode-1 unfoldings are $\widetilde{Q}^\top A_1$ and $Q^\top A_1$, and Step 4 gives $\|\widetilde{A} - A(Q,I,I)\|_F^2 \leq \epsilon^2 e_1$. The rank-$k$ tensor $B^\star(Q,I,I) = \sum_{i=1}^k (Q^\top u_i) \otimes v_i \otimes w_i$ satisfies $\|A(Q,I,I) - B^\star(Q,I,I)\|_F^2 \leq \mathrm{OPT}$, since $Q^\top$ is a contraction. *Magnitude and rounding of* $\widetilde{Q}^\top$. The matrix $S_0$ samples row $i$ of $Q$ with probability $p_i = \max\{\min\{1, \alpha\tau_i(Q)\}, n^{-2}\}$, where $\alpha = O(\epsilon^{-2}\log k)$, and rescales it by $p_i^{-1/2}$. The floor only increases the probabilities, and both Lemma B.13 and Lemma B.12 tolerate this. By Lemma B.12 with $m = O(\epsilon^{-2}k\log k)$ expected rows, we have $\|(S_0Q)^\dagger\|_{\mathrm{op}} \leq 2$ with probability at least $0.998$. Since every entry of $S_0$ is at most $p_i^{-1/2} \leq n$, on this event every entry of $\widetilde{Q}^\top = (S_0Q)^\dagger S_0$ has magnitude at most $2n$. We round the entries of $\widetilde{Q}^\top$ to $c\,n^\delta + \lceil \log_2(100/\epsilon)\rceil$ bits after the binary point, for a constant $c \geq 2$, before forming $\widetilde{A}$. By the hypotheses $\epsilon \geq 2^{-n^\delta}$ and $n^\delta \geq \log n$, the rounded entries and the entries of $\widetilde{A}'$ have $O(n^\delta)$ bits, and $\|\widetilde{A}'\|_F \leq 2^{O(n^\delta)}$. The rounding replaces $\widetilde{A}$ by $\widetilde{A}'$ with $\|\widetilde{A}' - \widetilde{A}\|_F \leq \sqrt{km}\cdot 2^{-c\,n^\delta}(\epsilon/100)\|A\|_F \leq \epsilon\sqrt{\mathrm{OPT}}/100$, where we use the hypothesis $\mathrm{OPT} \geq 2^{-n^\delta}\|A\|_F^2$ and $km \leq n^2 \leq 2^{2n^\delta}$. The exact basis $Q$ is likewise irrational in general, so the algorithm rounds its entries to $O(n^{\delta_2'})$ bits after the binary point, forms $S_0$ and $\widetilde{Q}^\top$ from the rounded $Q'$, and returns $U := Q'X_{j^\star}$. We write $\|X_{j^\star}\|_F\|V_{j^\star}\|_F\|W_{j^\star}\|_F \leq 2^{Cn^{\delta_2'}}$ for the outputs of the verifier, whose entries have $n^{\delta_2'}$ bits. Let $C'$ be the constant in the pseudo-inverse perturbation bound $\|(S_0Q')^\dagger S_0A_1 - (S_0Q)^\dagger S_0A_1\|_F \leq 2^{C'n^\delta}\|Q' - Q\|_F$, which holds on the subspace-embedding event $\sigma_{\min}(S_0Q) \geq 1/2$ of Lemma B.12. We choose the constant $c$ larger than $C+1$ and $C'+1$, which is possible since $C$ and $C'$ do not depend on $c$. Then, $\|Q' - Q\|_F \leq \sqrt{nk}\,2^{-cn^{\delta_2'}}$ perturbs the returned tensor and $\widetilde{Q}^\top A_1$ by at most $2^{-n^{\delta_2'}/2} \leq 0.01\epsilon\sqrt{\mathrm{OPT}}$, where we use $\log(nk) \leq n^\delta \leq n^{\delta_2'}$, $\epsilon \geq 2^{-n^\delta}$, and $\mathrm{OPT} \geq 2^{-n^\delta}\|A\|_F^2$. We charge both perturbations to the allowance $0.01\epsilon\sqrt{\mathrm{OPT}}$ of the rounding of $\widetilde{Q}^\top$, and we carry this allowance through the two triangle inequalities below. Therefore, using $e_1 \leq (1+\epsilon)^2\,\mathrm{OPT}$, we have

$$\|\widetilde{A}' - B^\star(Q,I,I)\|_F \leq \epsilon(1+\epsilon)\sqrt{\mathrm{OPT}} + \sqrt{\mathrm{OPT}} + 0.01\epsilon\sqrt{\mathrm{OPT}} \leq (1 + 1.62\epsilon)\sqrt{\mathrm{OPT}},$$

so the optimal rank-$k$ cost of $\widetilde{A}'$ is at most $(1+3.6\epsilon)\,\mathrm{OPT}$. Moreover, $B^\star(Q,I,I)$ is a rank-$k$ witness of this cost whose factors $Q^\top u_i, v_i, w_i$ have norms at most those of $U^\star, V^\star, W^\star$.

*The classical step.* We apply Lemma G.6 to $A' := \widetilde{A}'$ with the competitor $B' := B^\star(Q,I,I)$. Here $A'$ is a $k \times n \times n$ tensor with $O(n^\delta)$-bit entries, padded by zeros to $n \times n \times n$, so $\mathrm{nnz}(A') \leq kn^2$. The factor matrices of $B'$ are the padded $Q^\top U^\star$ and $V^\star, W^\star$, and their Frobenius norms are at most those of $U^\star, V^\star, W^\star$, and hence at most $2^{O(n^{\delta_2'})}$ by hypothesis. Each of the three runs returns $U_j', V_j, W_j \in \mathbb{R}^{n\times k}$, and we let $X_j \in \mathbb{R}^{k\times k}$ be the first $k$ rows of $U_j'$ and let $T_j := \sum_{i=1}^k (X_j)_i \otimes (V_j)_i \otimes (W_j)_i$. Dropping the other rows of $U_j'$ does not increase the cost, since the corresponding slabs of the

padded tensor are zero. For each run, the guarantee of Lemma G.6 holds with probability at least $9/10$, and in that case $\|\widetilde{A}' - T_j\|_F^2 \le (1+\epsilon)\|\widetilde{A}' - B^\star(Q,I,I)\|_F^2 + 2^{-\Omega(n^{\delta_2'})}\max\{1, \|\widetilde{A}'\|_F^2\} \le (1+\epsilon)(1+3.6\epsilon)\,\mathrm{OPT} + \epsilon\,\mathrm{OPT} \le (1+6\epsilon)\,\mathrm{OPT}$. We absorb the additive term as follows. Since the tensor $A \neq 0$ has $O(n^\delta)$-bit entries, we have $\|A\|_F^2 \ge 2^{-O(n^\delta)}$ and $\mathrm{OPT} \ge 2^{-n^\delta}\|A\|_F^2 \ge 2^{-O(n^\delta)}$. Moreover, $\|\widetilde{A}'\|_F^2 \le 2^{O(n^\delta)}$ and $\epsilon \ge 2^{-n^\delta}$. Therefore, $2^{-\Omega(n^{\delta_2'})}\max\{1, \|\widetilde{A}'\|_F^2\} \le 2^{-\Omega(n^{\delta_2'})+O(n^\delta)} \le \epsilon\,\mathrm{OPT}$, once the constant $C_0$ in $n^{\delta_2'} = C_0\, n^\delta 2^{O(k^2/\epsilon)}$ is taken large enough relative to the constants in the $O(n^\delta)$ bounds above. We do not use any lower bound on the competitor's own cost $\|\widetilde{A}' - B^\star(Q,I,I)\|_F^2$, and no such bound is available. This cost can vanish, e.g., when $Q$ spans exactly the column span of $U^\star$ and the residual is orthogonal to it.

*Selection.* With probability at least $1 - 10^{-3}$, at least one of the three independent runs succeeds. Let $c_j := \|\widetilde{A}' - T_j\|_F^2$ and $M_j := (\widetilde{A}' - T_j)_{(2)} \in \mathbb{R}^{n \times kn}$. Then, the estimate $\widehat{c}_j = \|GM_j\|_F^2$ has expectation $\|M_j\|_F^2 = c_j$ and variance at most $2c_j^2/s$, since each row $g$ of $\sqrt{s}\,G$ gives $\|g^\top M_j\|_2^2$ with mean $c_j$ and variance at most $2c_j^2$, and the $s$ rows are independent. Thus, by Chebyshev's inequality, $|\widehat{c}_j - c_j| \le (\epsilon/4)c_j$ fails for a given $j$ with probability at most $32/(s\epsilon^2)$, and hence it holds for all three $j$ with probability at least $1 - 2\cdot 10^{-3}$ once $s \ge 48000\,\epsilon^{-2}$. On these events, the selected candidate $T := T_{j^\star}$ satisfies $c_{j^\star} \le \frac{1+\epsilon/4}{1-\epsilon/4}\min_j c_j \le (1+\epsilon)(1+6\epsilon)\,\mathrm{OPT} \le (1+7.7\epsilon)\,\mathrm{OPT}$ for $\epsilon \le 0.1$. Here $X := X_{j^\star}$, $V := V_{j^\star}$, $W := W_{j^\star}$ are the matrices returned by the algorithm. By the triangle inequality once more, we have

$$\begin{aligned}\|A(Q,I,I) - T\|_F &\le \|A(Q,I,I) - \widetilde{A}\|_F + \|\widetilde{A} - \widetilde{A}'\|_F + \|\widetilde{A}' - T\|_F \\ &\le \big(\epsilon(1+\epsilon) + 0.01\epsilon + \sqrt{1+7.7\epsilon}\big)\sqrt{\mathrm{OPT}},\end{aligned}$$

hence $\|A(Q,I,I) - T\|_F^2 \le (1+11\epsilon)\,\mathrm{OPT}$ for $\epsilon \le 0.1$.

*Lifting.* The output $U \otimes V \otimes W$ with $U = Q'X$ is the tensor $\sum_{i=1}^k (Q'X_i) \otimes V_i \otimes W_i$ of rank at most $k$, and its mode-1 unfolding is $Q' \cdot T_{(1)}$, where $T_{(1)}$ is the mode-1 unfolding of $T$. By Step 2 with $Y = T_{(1)}$, we have

$$\begin{aligned}\|QT_{(1)} - A_1\|_F^2 = e_1 + \|Q^\top A_1 - T_{(1)}\|_F^2 &= e_1 + \|A(Q,I,I) - T\|_F^2 \\ &\le (1+\epsilon)^2\,\mathrm{OPT} + (1+11\epsilon)\,\mathrm{OPT} \le (2+13.1\epsilon)\,\mathrm{OPT}\end{aligned}$$

for $\epsilon \le 0.1$. By the rounding of $Q$ above, $\|(Q' - Q)T_{(1)}\|_F \le 0.01\epsilon\sqrt{\mathrm{OPT}}$, and hence, by the triangle inequality, we have $\|U \otimes V \otimes W - A\|_F^2 \le (2+13.1\epsilon)\,\mathrm{OPT} + 0.04\epsilon\,\mathrm{OPT} \le (2+14\epsilon)\,\mathrm{OPT}$. The failure events are the coreset call at $0.001$, the applications of Lemma B.13 and Lemma B.12 to $S_0$ at $0.004$ and $0.002$, respectively, by Remark B.14, the event of probability $10^{-3}$ that all three classical runs fail, and the three cost estimates at $2\cdot 10^{-3}$. By a union bound, the success probability is at least $0.99$.

*Running time and queries.* Computing $C_1$ takes $\widetilde{O}(\epsilon^{-1}n^2k^{0.5} + nk^{\omega-1})$ time and $\widetilde{O}(\epsilon^{-1}nk^{0.5})$ mode-1 fiber queries, and as in Theorem G.4, computing $Q$, $S_0$, $\widetilde{Q}^\top$ costs $\widetilde{O}(\epsilon^{2-2\omega}nk^{\omega-1}) + \mathrm{poly}(k/\epsilon)$. Forming $\widetilde{A}$ reads the $m$ slabs $A_{i,*,*}$, $i \in \mathrm{supp}(S_0)$, i.e., $mn = \widetilde{O}(\epsilon^{-2}nk)$ mode-3 fibers, and computes $kn^2$ entries at $O(m)$ arithmetic operations each, which takes $O(n^2km) = \widetilde{O}(\epsilon^{-2}n^2k^2)$ time. Each of the three classical runs takes $(\mathrm{nnz}(\widetilde{A}) + n\,\mathrm{poly}(k/\epsilon) + 2^{O(k^2/\epsilon)})n^{O(\delta)} = (kn^2 + n\,\mathrm{poly}(k/\epsilon) + 2^{O(k^2/\epsilon)})n^{O(\delta)}$ time. Each cost estimate $\widehat{c}_j$ is computed exactly as $\|G\widetilde{A}'_{(2)} - (GV_j)Z_j\|_F^2$, where $Z_j \in \mathbb{R}^{k\times kn}$ is the matrix whose $i$-th row is $\mathrm{vec}((X_j)_i(W_j)_i^\top)$. Here $\widetilde{A}'$ is the tensor written as $\widetilde{A}$ in Algorithm 11, and we factor the scalar $s^{-1/2}$ out of $G$, so all intermediate quantities are fixed-point numbers. The product $G\widetilde{A}'_{(2)}$ costs $O(skn^2) = O(\epsilon^{-2}kn^2)$ operations on $O(n^\delta)$-bit numbers. Computing $GV_j$, $(GV_j)Z_j$, and the final norm costs $O(\epsilon^{-2}nk^2)$ operations on $n^{\delta_2'}$-bit numbers, each $\mathrm{poly}(n^{\delta_2'}) = n^{O(\delta)}2^{O(k^2/\epsilon)}$ word operations. The second of these takes $ab \cdot n^{O(\delta)}$ time with $a = \epsilon^{-1}nk$ and $b = \epsilon^{-1}k\,2^{O(k^2/\epsilon)}$, which is at most $(a^2+b^2)\cdot n^{O(\delta)} = (\epsilon^{-2}n^2k^2 + 2^{O(k^2/\epsilon)})\cdot n^{O(\delta)}$, since $b^2 = 2^{O(k^2/\epsilon)}$ due to $\log(k/\epsilon) = O(k^2/\epsilon)$. Finally, computing $U = Q'X_{j^\star}$ takes $O(nk^2)$ time, and summing up these costs gives the stated bounds. □

**Algorithm 12** Converting a tensor low-rank approximation to a CURT decomposition.

**procedure** FROMLOWRANKTOCURT($A, U_B, V_B, W_B, n, k, \epsilon$) ▷ Lemma G.8
  $d_1 \leftarrow d_2 \leftarrow d_3 \leftarrow O(k \log k + k/\epsilon)$.
  $\epsilon_0 \leftarrow 0.01$.
  Let $B_1 := V_B^\top \odot W_B^\top \in \mathbb{R}^{k \times n^2}$ (implicit, and only $B_1 D_1$ is formed, in $O(kd_1)$ time)
  $D_1 \leftarrow$TENSORLEVERAGESCORE($V_B^\top, W_B^\top, n, n, k, \epsilon_0, d_1$)
  Form $\widehat{U} = A_1 D_1 (B_1 D_1)^\dagger \in \mathbb{R}^{n \times k}$.
  Let $B_2 := \widehat{U}^\top \odot W_B^\top \in \mathbb{R}^{k \times n^2}$ (implicit, and only $B_2 D_2$ is formed, in $O(kd_2)$ time)
  $D_2 \leftarrow$TENSORLEVERAGESCORE($\widehat{U}^\top, W_B^\top, n, n, k, \epsilon_0, d_2$).
  Form $\widehat{V} = A_2 D_2 (B_2 D_2)^\dagger \in \mathbb{R}^{n \times k}$
  Let $B_3 := \widehat{U}^\top \odot \widehat{V}^\top \in \mathbb{R}^{k \times n^2}$ (implicit, and only $B_3 D_3$ is formed, in $O(kd_3)$ time)
  $D_3 \leftarrow$TENSORLEVERAGESCORE($\widehat{U}^\top, \widehat{V}^\top, n, n, k, \epsilon_0, d_3$)
  $C \leftarrow A_1 D_1$, $R \leftarrow A_2 D_2$, $T \leftarrow A_3 D_3$ ▷ $D_i = \widetilde{D}_i \Sigma_i$: selection $\widetilde{D}_i$, weights $\Sigma_i$
  $U \leftarrow \sum_{i=1}^{k} ((B_1 D_1)^\dagger)_i \otimes ((B_2 D_2)^\dagger)_i \otimes ((B_3 D_3)^\dagger)_i$
  **return** $C$, $R$, $T$ and $U$
**end procedure**

### G.4 Tensor CURT Decomposition: Fixed-Parameter Tractable and Bicriteria

*Note on Algorithm 12.* Line 12: the returned factors are the unweighted selections $C = A_1 \widetilde{D}_1$, $R = A_2 \widetilde{D}_2$, $T = A_3 \widetilde{D}_3$, and the weights $(\Sigma_1)_{ii} (\Sigma_2)_{jj} (\Sigma_3)_{ll}$ are folded into $U_{ijl}$.

**Lemma G.8** (A modification of Theorem C.40 in Song et al. [2019])**.** *Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$, let $k \geq 1$, and let $U_B, V_B, W_B \in \mathbb{R}^{n \times k}$ be any three matrices (a competitor). Then there is a classical algorithm (Algorithm 12) which takes $\widetilde{O}(n \operatorname{poly}(k/\epsilon))$ time and outputs three matrices $C \in \mathbb{R}^{n \times c}$ with columns from A, $R \in \mathbb{R}^{n \times r}$ with rows from A, $T \in \mathbb{R}^{n \times t}$ with tubes from A, and a tensor $U \in \mathbb{R}^{c \times r \times t}$ of CP rank at most k such that $c = r = t = O(k \log k + k/\epsilon)$, and*

$$\left\| \sum_{i=1}^{c} \sum_{j=1}^{r} \sum_{l=1}^{t} U_{i,j,l} \cdot C_i \otimes R_j \otimes T_l - A \right\|_F^2 \leq (1+\epsilon) \left\| \sum_{i=1}^{k} (U_B)_i \otimes (V_B)_i \otimes (W_B)_i - A \right\|_F^2$$

*holds with probability at least* $0.99$ *(three applications of Lemma B.13, each run with failure probability* $0.002$ *at a constant factor in $c, r, t$ by Remark B.14, and three* TENSORLEVERAGESCORE *calls with internal sketch failure* $0.001$ *each).*

The proof of Theorem C.40 of Song et al. [2019] shows this competitor-relative form, which is the form we need below, and we follow it with only three changes. We form $A_i D_i$ by reading the $d_i$ sampled fibers in $O(nd_i)$ time, in place of the $\operatorname{nnz}(A)$ pass. The sampler of Lemma G.1 contributes the polylogarithmic factor. The three regressions are unconstrained $k$-row regressions, and we solve them by argmin on leverage-score column samples of their own designs, so Lemma B.13 applies within its hypotheses.

**Theorem G.9** (Bicriteria Tensor CURT Decomposition)**.** *Given a 3rd order tensor $A \in \mathbb{R}^{n \times n \times n}$ and a positive integer $k \leq n$, $\epsilon \in (0, 0.1)$, there exists an algorithm which takes $\widetilde{O}(\epsilon^{-1} n^2 k^{0.5} + n^{1.5} \operatorname{poly}(k/\epsilon))$ time, $\widetilde{O}(\epsilon^{-1} n k^{0.5} + \epsilon^{-3} n^{0.5} k^2 + \operatorname{poly}(k/\epsilon))$ queries to the rows, columns and tubes of A (the additive term reads the sampled fibers of the conversion) and outputs three matrices $C, R, T \in \mathbb{R}^{n \times m}$ with $m = O(r \log r + r/\epsilon) = \widetilde{O}(k^2/\epsilon)$, where $r = k^2$ bounds the CP rank of the bicriteria solution of Theorem G.4, and $U \in \mathbb{R}^{m \times m \times m}$ of CP rank at most $r$ such that*

$$\left\| \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{l=1}^{m} U_{i,j,l} \cdot C_i \otimes R_j \otimes T_l - A \right\|_F^2 \leq (2+\epsilon) \cdot \inf_{\text{rank-}k\ A_k} \|A - A_k\|_F^2$$

*with probability at least* $0.89$.

*Proof.* It directly follows from combining Theorem G.4 and Lemma G.8. We invoke Lemma G.8 with rank parameter $r = k^2$ and the bicriteria solution $B = \sum_{i,t \in [k]} Q_t \otimes V_i \otimes y_{i,t}$ as the competitor,

whose factor matrices are the $n \times k^2$ matrices with columns $Q_t$, $V_i$, and $y_{i,t}$ indexed by the pairs $(i,t)$. The three fiber counts of the lemma are then all $O(r \log r + r/\epsilon) = m$. Moreover, the lemma's $\widetilde{O}(n \operatorname{poly}(k/\epsilon))$ time and $\operatorname{poly}(k/\epsilon)$ fiber reads are absorbed, and the two failure probabilities add to at most $0.11$. Finally, the product $(1+\epsilon)(2+\epsilon)$ of the two guarantees is $(2+\epsilon)$ after rescaling $\epsilon$ by a constant. □

**Theorem G.10** (Fixed-Parameter Tractable Tensor CURT Decomposition)**.** *Let $\epsilon \in (0, 0.1)$, and let $A \in \mathbb{R}^{n \times n \times n}$, $k$, $\delta$ and* OPT *satisfy the hypotheses of Theorem G.7 with $\epsilon/4$ in place of $\epsilon$, in particular $\epsilon \geq 2^{2-n^{\delta}}$. There is a quantum algorithm that outputs $C \in \mathbb{R}^{n \times c}$, $R \in \mathbb{R}^{n \times r}$ and $T \in \mathbb{R}^{n \times t}$ with $c = r = t = O(k \log k + k/\epsilon)$, consisting of columns, rows and tubes of $A$, and $U \in \mathbb{R}^{c \times r \times t}$ of CP rank at most $k$, such that*

$$\Big\| \sum_{i=1}^{c} \sum_{j=1}^{r} \sum_{l=1}^{t} U_{i,j,l}\, C_i \otimes R_j \otimes T_l - A \Big\|_F^2 \leq (2+\epsilon) \cdot \mathrm{OPT}$$

*with probability at least $0.98$, where* OPT *is the infimum of Theorem G.7. The algorithm runs in $(n^2k^2/\epsilon^2 + n \operatorname{poly}(k/\epsilon) + 2^{O(k^2/\epsilon)}) \cdot n^{O(\delta)}$ time in the unit-cost RAM model with $O(\log n)$-bit words of Theorem G.7, and uses $\widetilde{O}(\epsilon^{-1} n k^{0.5})$ queries to mode-1 fibers, $\widetilde{O}(\epsilon^{-2} n k)$ queries to mode-3 fibers and $\operatorname{poly}(k/\epsilon)$ further queries to the rows, columns and tubes of $A$.*

*Proof.* We run Algorithm 11 with accuracy $\epsilon/4$, and then feed its output $U, V, W \in \mathbb{R}^{n \times k}$ to Algorithm 12, also run with accuracy $\epsilon/4$, as the competitor of Lemma G.8. Note that the lemma holds for every fixed competitor, and the randomness of the conversion is independent of $U, V, W$. Therefore, the output has cost at most $(1+\epsilon/4)(2+\epsilon/4) \cdot \mathrm{OPT} = (2 + \frac{3}{4}\epsilon + \epsilon^2/16) \cdot \mathrm{OPT} \leq (2+\epsilon) \cdot \mathrm{OPT}$, since $\epsilon^2/16 \leq \epsilon/4$ for $\epsilon \leq 4$. The fiber counts are $O(k \log k + 4k/\epsilon) = O(k \log k + k/\epsilon)$, and the two failure probabilities add to at most $0.02$. Theorem G.7 at accuracy $\epsilon/4$ contributes its stated time and query bounds with $\epsilon/4$ in place of $\epsilon$, and this substitution leaves each term in its stated $O(\cdot)$, $\operatorname{poly}(\cdot)$, or $\widetilde{O}(\cdot)$ class. The conversion performs $\widetilde{O}(n \operatorname{poly}(k/\epsilon))$ arithmetic operations on numbers of $\operatorname{poly}(k)\, n^{\delta_2'} = \operatorname{poly}(n^{\delta_2'})$ bits, since the inputs have $O(n^{\delta_2'})$ and $O(n^{\delta})$ bits and the pseudo-inverses grow the bit size by $\operatorname{poly}(k)$ factors, and each operation costs $\operatorname{poly}(n^{\delta_2'})$ word operations. By the AM–GM inequality, we have $n \operatorname{poly}(k/\epsilon) \operatorname{poly}(n^{\delta_2'}) \leq n^2 + \operatorname{poly}(k/\epsilon)^2 \operatorname{poly}(n^{\delta_2'})^2$, and using $n^{\delta_2'} = O(n^{\delta} 2^{O(k^2/\epsilon)})$ and $\log(k/\epsilon) = O(k^2/\epsilon)$, the right-hand side is $(n^2 + 2^{O(k^2/\epsilon)}) \cdot n^{O(\delta)}$, which, with the polylogarithmic factor absorbed in $n^{O(\delta)}$ since $n^{\delta} \geq \log n$, is within the stated time. Finally, the conversion reads $d_1 + d_2 + d_3 = O(k \log k + k/\epsilon)$ sampled fibers of $A$, which accounts for the additive query term. □

# H Improved Quantum Coreset Algorithm for $(k,p)$-Clustering and Application

In this section, we give an improved quantum coreset construction for $(k,p)$-clustering. We observe that the size of the coreset obtained in the main result of Xue et al. [2023] (their Theorem 3.1) scales linearly with $d$, which causes an additional $d^{0.5}$ factor in the final runtime.

We begin by recalling the $(k,p)$-clustering problem in $\mathbb{R}^d$: let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$, $X = (\mathbb{R}^d)^k$ and $\operatorname{cost}(a_i, x) = \min_{j \in [k]} \|a_i - x_j\|_2^p$, where $p \geq 1$ is the power of the distance, and $x_j$ is one of the centers in $x$. When $p = 1$, this is the well-studied $k$-median problem, and when $p = 2$, this captures the $k$-means problem. To construct a coreset, we implement the sensitivity sampling framework in quantum sublinear time.

We need the following quantum algorithm, due to Xue et al. [2023], that computes an $(\alpha, \beta)$-bicriteria approximation.

**Definition H.1** (Bicriteria Approximation)**.** *Let $A \subset \mathbb{R}^d$, assume* OPT *is the optimal cost of the $(k,p)$-clustering problem for $A$, we say a set $x \subset \mathbb{R}^d$ is an $(\alpha, \beta)$-bicriteria approximation if $|x| \leq \alpha k$ and $\operatorname{cost}(A, x) \leq \beta\, \mathrm{OPT}$.*

**Lemma H.2** (Lemma 3.7 of Xue et al. [2023], with a size cap)**.** *Let $A \subset \mathbb{R}^d$ with $n \geq 2$ points, there exists a quantum algorithm that outputs a set $x$ with at most $c_{\mathrm{bi}} k \lceil \log_2 n \rceil^2$ points on every*

*outcome, where* $c_{\mathrm{bi}} := 354$*, such that* $x$ *is an* $(O(\log^2 n), 2^{O(p)})$*-bicriteria approximation to the* $(k,p)$*-clustering problem for* $A$ *with probability at least* $5/6$*. The algorithm uses* $\widetilde{O}(\sqrt{nk})$ *queries to* $A$*, takes* $\widetilde{O}(\sqrt{nk}d)$ *time and* $\mathrm{poly}(k \log n)$ *additional preprocessing time.*

Note that $\mathrm{cost}(A, x)$ does not increase when centers are added to $x$.

The size cap is our modification of the algorithm of Xue et al. [2023], and Lemma H.13 needs it. Their logarithms are to base 2, as their Appendix A uses that $\lceil \log n \rceil$ halvings reduce $n$ points to at most one, and we write them as $\log_2$. Their Algorithm 1 takes the union of three repetitions of the following procedure. A loop adds a sample of $13k\lceil \log_2 n \rceil$ points to the output in each iteration and shrinks a set $D_t \subseteq A$ of remaining points. The loop stops once an estimate $\widetilde{r}_t$ of $|D_t|$ with $\left|\widetilde{r}_t - |D_t|\right| \le |D_t|/2$ (their Line 8) satisfies $\widetilde{r}_t \le 39k\lceil \log_2 n \rceil$, or after $3\lceil \log_2 n \rceil$ iterations, and then their Line 11 adds all points of $D_t$ to the output. The proof in their Appendix A shows that, with probability at least $5/6$, one of the three repetitions has cost at most $2^{O(p)}\,\mathrm{OPT}$, and that this repetition stops its loop by the first condition with all subroutines successful, so that $|D_t| \le 2\widetilde{r}_t \le 78k\lceil \log_2 n \rceil$ for it. In the other repetitions, the loop may stop by the second condition, which we call a late stop, or with a wrong estimate, and then $D_t$ can contain almost all of $A$. Hence, their output has no worst-case size bound. We make two changes. First, we keep at most $13k\lceil \log_2 n \rceil$ points of each sample, which changes nothing when their sampler succeeds. Second, we implement their Line 11 by the repeated Grover search described after Lemma B.15, with $q_i = 1$ on $D_t$ and $q_i = 0$ elsewhere, computed by their membership oracle of $D_t$, and with the failure probability $1/\mathrm{poly}(nk)$ of their subroutines. This search finds the points of $D_t$ one at a time, and we stop it as soon as it has found $78k\lceil \log_2 n \rceil + 1$ points. Since $\lceil \log_2 n \rceil \ge 1$ for $n \ge 2$, each repetition then outputs at most $3\lceil \log_2 n \rceil \cdot 13k\lceil \log_2 n \rceil + 78k\lceil \log_2 n \rceil + 1 \le 118k\lceil \log_2 n \rceil^2$ points, and each run outputs at most $3 \cdot 118k\lceil \log_2 n \rceil^2 = c_{\mathrm{bi}} k \lceil \log_2 n \rceil^2$ points. The stop does not change the output of the repetition used by their proof, since $|D_t| \le 78k\lceil \log_2 n \rceil$ there, and by the monotonicity above, the points of the other repetitions do not increase the cost of the union. Therefore, the success probability is still at least $5/6$. The stop also bounds the expected cost of their Line 11 by $\widetilde{O}(\sqrt{nk})$ membership queries on every outcome of the loop, while without it this bound holds only when $|D_t| = \widetilde{O}(k)$. Indeed, let $J \le 78k\lceil \log_2 n \rceil + 1$ be the number of searches that find a point. For $j \le J$, the $j$-th search starts with $|D_t| - j + 1 \ge J - j + 1$ points of $D_t$ not yet found and takes $O(\sqrt{n/(|D_t| - j + 1)})$ queries in expectation. Hence, these $J$ searches take $O(\sum_{i=1}^{J} \sqrt{n/i}) = O(\sqrt{nk \log n})$ queries in expectation, and at most one final search that finds no point takes $\widetilde{O}(\sqrt{n})$ queries.

We also need a quantum approximate nearest neighbor oracle, which would be crucial to approximately find the center a point belongs to.

**Lemma H.3** (Lemma 3.4 of Xue et al. [2023])**.** *Let* $A \subset \mathbb{R}^d$ *and* $x \subset \mathbb{R}^d$ *with* $|x| = m$*, given two parameters* $\delta > 0$*,* $c_\tau \in [2.5, 3)$*, there exists a quantum oracle that given* $a_i \in A$*, returns* $\tau(a_i) \in x$*, using* $\mathrm{poly}(m \log(n/\delta))$ *preprocessing time. With probability at least* $1 - \delta$*,* $\tau : A \to x$ *is a mapping such that for all* $a_i \in A$*,*

$$\|a_i - \tau(a_i)\|_2 \le c_\tau \cdot \mathrm{dist}(a_i, x), \quad \textit{hence} \quad \|a_i - \tau(a_i)\|_2^p \le c_\tau^p \cdot \mathrm{cost}(a_i, x) < 3^p \,\mathrm{cost}(a_i, x).$$

*Each query to* $\tau$ *takes* $O(d\,\mathrm{poly}\log(mn/\delta))$ *time.*

Note that $\tau$ is also a *partition oracle*, as we could assign $a_i$ to $\tau(a_i)$, which is one of the $m$ clusters.

We need two other ingredients: one being estimating $\mathrm{cost}(A, x) = \sum_{i=1}^{n} \mathrm{cost}(a_i, x)$ and the other being estimating the number of points falling in each cluster.

**Lemma H.4** (Lemma 6 of Li et al. [2019], with quantum maximum finding [Dürr and Høyer, 1996])**.** *Let* $C = \{c_1, \ldots, c_n\}$ *be a collection of nonnegative numbers given by an oracle, let* $c = \sum_{i=1}^{n} c_i$*, and let* $\epsilon, \delta \in (0, 1)$*. Suppose a number* $M > 0$ *is known with* $c_i \in [0, M]$ *for all* $i \in [n]$*, and each* $c_i/M$ *is specified with* $O(\log(n/\epsilon))$ *bits. There exists a quantum algorithm that outputs an approximation* $\widetilde{c}$ *where* $\widetilde{c} = (1 \pm \epsilon) \cdot c$ *with probability at least* $1 - \delta$*, using* $\widetilde{O}(\sqrt{n}\log(1/\delta)/\epsilon)$ *queries to* $C$*. The same bounds hold for arbitrary nonnegative reals* $c_i$ *if* $M := \max_{i \in [n]} c_i$ *is supplied, or is computed by Lemma H.11 below on the values* $-c_i$ *with* $\widetilde{O}(\sqrt{n}\log(1/\delta))$ *additional queries and each of the two steps run with failure probability* $\delta/2$*.*

Lemma 6 of Li et al. [2019] estimates the mean of values $F(i) \in [0, 1]$ given with $l$ bits of precision within a factor $1 \pm \epsilon$ with probability $2/3$ using $O(\sqrt{n}/\epsilon)$ queries, and its extra factor $l \log l$ is

absorbed in $\widetilde{O}$. We apply it to the values $c_i/M$ and take the median of $O(\log(1/\delta))$ repetitions. When $M = \max_i c_i$, we have $M \le c$, so truncating each $c_i/M$ to $l = \lceil \log_2(2n/\epsilon) \rceil$ bits changes $c$ by at most $n2^{-l}M \le \epsilon c/2$, which is absorbed by rescaling $\epsilon$. Moreover, $M = 0$ forces $c = 0$.

**Lemma H.5** (Theorem 4.4 of Xue et al. [2023]). *Let $A \in (\mathbb{R}^d)^n, x \in (\mathbb{R}^d)^m$ and $\tau : A \to x$, let $C_j = \{a \in A : \tau(a) = x_j\}$, let $\epsilon \in (0, 1/3), \delta > 0$, then there exists a quantum algorithm that with probability at least $1 - \delta$, outputs a list of estimates $\widetilde{n}_j$ for all $j \in [m]$ where $\widetilde{n}_j = (1 \pm \epsilon) \cdot |C_j|$, using $\widetilde{O}(\sqrt{nm/\epsilon} \log(1/\delta))$ queries to $\tau$ and an additional $\widetilde{O}((\sqrt{nm/\epsilon} + m/\epsilon) \log(n/\delta))$ time.*

**Lemma H.6** (Theorem 4.2 of Xue et al. [2023], multidimensional approximate summation). *Let $1 \le m \le n$ be integers, $\tau : A \to [m]$, and let $f : A \to \{0, 1, \ldots, M\}$ be integer-valued, both computable by one oracle call per point. Let $\epsilon \in (0, 1/3)$ and $\delta > 0$. There exists a quantum algorithm that, with probability at least $1 - \delta$, outputs estimates $\widetilde{s}_j = (1 \pm \epsilon) \cdot \sum_{a \in A : \tau(a) = j} f(a)$ for all $j \in [m]$ simultaneously, using $\widetilde{O}(\sqrt{nm/\epsilon} \log(1/\delta) \log M)$ queries and $\widetilde{O}((\sqrt{nm/\epsilon} + m/\epsilon) \log(n/\delta) \log M)$ additional time. (Their proof applies Lemma H.5 to each of the $O(\log M)$ binary digits of $f$, which gives relative accuracy on every cell, however light.)*

Our algorithm, given in Section H.1, is a quantum implementation of the one-shot coreset construction of Huang et al. [2024]. It has two main advantages: (1) Seeded with a discrete constant-factor solution, it computes a coreset with points only from $A$. (2) Its sampling probabilities are relatively easy to compute.

**Remark H.7** (Black-box use of a one-shot sampler). *Our Grover-based sampling step uses a one-shot sampler only through its sampling probabilities. Given* any *one-shot clustering sensitivity sampler that, after computing approximate centers, produces a coreset of size $s$ and evaluates one sampling probability in time $\mathcal{T}$, the procedure of Lemma B.15 performs the sampling step in $\widetilde{O}(\sqrt{ns}) \cdot \mathcal{T}$ time. Consequently, any improvement to classical one-shot clustering coresets, in the size $s$ or the evaluation time $\mathcal{T}$, immediately improves this step of our quantum algorithm.*

### H.1 One-Shot Sensitivity Sampling in Sublinear Time

By Remark H.7, the sampling step of a one-shot sampler costs $\widetilde{O}(\sqrt{ns})$ evaluations of its sampling probabilities. In this section, we show that the cell statistics that a one-shot sampler needs can be estimated quantumly, and we prove that the coreset construction of Huang et al. [2024] can be implemented in this way, with the same coreset size up to polylogarithmic factors, yielding Theorem H.15.

**Corollary H.8** (Cell statistics estimation). *Let $c : A \to [M]$ be a partition oracle computable in time $\mathcal{T}_{\text{cell}}$ per point, and let $v : A \to \mathbb{R}_{\ge 0}$ be a value function computable together with $c(a)$ in the same time, for instance $v(a) = \operatorname{dist}^p(a, x')$ for a center set $x'$. For each cell $c \in [M]$, let $n_c := |\{a \in A : c(a) = c\}|$ and $\sigma_c := \sum_{a \in A : c(a) = c} v(a)$. Suppose that the values satisfy $v(a) \in \{0\} \cup [F, RF]$ for all $a \in A$, for known $F > 0$ and $R \ge 1$. Then for any $\delta > 0$, with probability at least $1 - \delta$, all $n_c$ and $\sigma_c$ can be estimated within a factor $1 \pm 1/4$ using $\widetilde{O}(\sqrt{nM} \log(2R) \log(1/\delta))$ queries and $\widetilde{O}(\sqrt{nM} \cdot \mathcal{T}_{\text{cell}} \log(2R) \log(1/\delta))$ time.*

*Proof.* The masses are estimated by Lemma H.5 with the partition oracle $c$ in place of $\tau$ and accuracy $1/4$, using $\widetilde{O}(\sqrt{nM})$ queries. For the cost sums, let $u := F/16$ and $f(a) := \lfloor v(a)/u \rceil \in \{0, 1, \ldots, \lceil 16R \rceil\}$, where we round to the nearest integer. If $v(a) = 0$, then $f(a) = 0$, and if $v(a) \ge F$, the rounding changes $v(a)$ by at most $u/2 \le v(a)/32$. Therefore, $u \sum_{a : c(a) = c} f(a) \in [\frac{15}{16}\sigma_c, \frac{17}{16}\sigma_c]$ for every cell. By Lemma H.6 with accuracy $1/8$ and $\log\lceil 16R \rceil = O(\log(2R))$, we can estimate all the integer sums within $1 \pm 1/8$ in $\widetilde{O}(\sqrt{nM} \log(2R) \log(1/\delta))$ queries. Note that $(1 + 1/8)(17/16) < 5/4$ and $(1 - 1/8)(15/16) > 3/4$. □

**Remark H.9** (Instantiation with the construction of Huang et al. [2024]). *We verified the construction of Huang et al. [2024] against their full version. Their Algorithm 1 performs a* single *round of importance sampling. Given an $O(1)$-approximate solution $A^\star$ with $k$ centers, each cluster is partitioned into rings by the ratio of $\operatorname{dist}^p(a, A^\star)$ to the cluster's average cost $\Delta_i$. The rings are merged by cost into $O_p(\log^2(k/\epsilon))$ groups, as in their Lemma A.5, and each group $G$ contributes $\Gamma_G$ samples drawn with probability proportional to $\operatorname{dist}^p(a, A^\star)$ within $G$. Their algorithm runs in $O(nk)$ classical time. Two adjustments are needed. First, their coreset is* not *points-only as written,*

*since Lines 2–3 of their Algorithm 1 add the centers $A^\star$ with weights $|P_i \setminus \mathcal{G}|$, which represent the discarded inner and light rings. We repair this by taking $A^\star$ to be a* discrete *constant-factor approximation, i.e., one whose centers are input points, and their algorithm permits this at the cost of a $2^{O(p)}$ factor in the approximation constant. Second, their sampler draws $\Gamma_G$ points* with replacement *within each group, while our primitive performs independent per-point sampling, and converting to independent sampling with $q_a = \min\{1, \Gamma_G \cdot \mathrm{dist}^p(a, A^\star)/\mathrm{cost}(G, A^\star)\}$ is a standard Poissonization step. Lemma H.10 below shows that their analysis tolerates both adjustments, together with the constant-factor statistics estimates and the approximate center weights provided by our quantum access.*

**Lemma H.10** (Robustness of the one-shot sampler of Huang et al. [2024])**.** *Let $p \geq 1$ be a constant, $A \subset \mathbb{R}^d$, and let $A^\star$ be a $2^{O(p)}$-approximate solution with induced clusters $P_1, \ldots, P_k$ and average costs $\Delta_i$. Suppose that the ring and group decomposition of Huang et al. [2024] is constructed from estimates $\widehat{\Delta}_i \in [\Delta_i/2, 2\Delta_i]$ and ring-cost estimates correct up to a factor of 2, that each group $G$ is sampled by independent Bernoulli variables with probabilities $q_a = \min\{1, \Gamma_G \cdot \mathrm{dist}^p(a, A^\star)/\widehat{c}_G\}$ and weights $1/q_a$, where $\widehat{c}_G \in [\mathrm{cost}(G, A^\star)/2,\ 2\,\mathrm{cost}(G, A^\star)]$, and that each center $a_i^\star$ receives an estimated weight $\widehat{w}(a_i^\star) = |P_i \setminus \mathcal{G}| \pm \epsilon|P_i|$. If the sample budgets $\Gamma_G$ are as in Theorem A.11 of Huang et al. [2024], inflated by a factor of $2^{O(p)}$, then with probability at least $0.99$ the output is an $O(\epsilon)$-coreset of $A$ of size $k + 2^{O(p)} \sum_G \Gamma_G$. Moreover, for every dimension $d$ and every integer $n \geq \max\{2, |A|, k\}$, and without any preprocessing of the input, the budgets may be taken as $\Gamma_G = O_p(k^{\frac{2p+2}{p+2}} \epsilon^{-2} \log n \cdot \log^4(2dk/\epsilon))$ for the main groups and $\Gamma_G = O_p(k\epsilon^{-2} \log n \cdot \log^3(2dk/\epsilon))$ for the outer groups, so the coreset has size $\widetilde{O}_p(k^{\frac{2p+2}{p+2}} \epsilon^{-2})$. Finally, for an* arbitrary *center set $A^\star \in (\mathbb{R}^d)^k$ with induced clusters $P_1, \ldots, P_k$ and the same estimates and budgets, the output $(S, w)$ satisfies, with probability at least $0.99$, $|\sum_{s \in S} w_s \,\mathrm{dist}^p(s, C) - \mathrm{cost}(A, C)| \leq O(\epsilon) \cdot (\mathrm{cost}(A, C) + \mathrm{cost}(A, A^\star))$ for all $C \in (\mathbb{R}^d)^k$. On the same event, the output has size $k + 2^{O(p)} \sum_G \Gamma_G$.*

*Proof.* We reduce to the analysis of Theorem A.11 of Huang et al. [2024], given in their Appendices A.5–A.7, and we track every place where that analysis interacts with the three modifications.

*Step 1: the perturbed decomposition.* The constructed rings $\widehat{R}_{ij}$ partition each $P_i$, and membership gives the two-sided bound $2^j \widehat{\Delta}_i \leq \mathrm{dist}^p(a, A^\star) < 2^{j+1} \widehat{\Delta}_i$. Every object of their analysis is defined relative to the decomposition the algorithm actually built. These objects include the huge and tiny subsets $H(G, C)$ and $T(G, C)$, the far subsets $F(G, C)$, the cost vectors of their Definition A.14, and the coverings, so we run their argument with respect to the hatted decomposition throughout. First, their Eq. (7) and Observation A.7 convert ring membership into distance bounds, and they hold verbatim with $\widehat{\Delta}_i$ in place of $\Delta_i$. Second, rings placed in a common group have true costs within a factor of 8 of one another instead of 2, so the constants of their Observation A.4 inflate from $(2k, 4k)$ to $(8k, 32k)$. Third, the covering-number bounds are unaffected, because their Lemma B.7 is applied with radii $r_i = 2^{j+1}\widehat{\Delta}_i$ and their Definition B.8 partitions by the ratio $\mathrm{dist}^p(a_i^\star, C)/\widehat{\Delta}_i$ with the same number of levels. Finally, the identity $|P_i| \cdot \Delta_i = \mathrm{cost}(P_i, A^\star)$ is used only to bound the total cost of inner rings and of discarded light groups, and there $\widehat{\Delta}_i \leq 2\Delta_i$ loses a factor of 2. Each effect inflates constants by at most $2^{O(p)}$, which is absorbed by rescaling $\epsilon$.

*Step 2: Bernoulli sampling.* We write the estimator of a cost vector $y$ as $\sum_{a \in G} \xi_a y_a/q_a$ with $\xi_a \sim \mathrm{Ber}(q_a)$ independent. This estimator is exactly unbiased regardless of the accuracy of $\widehat{c}_G$, because each weight is the reciprocal of the probability actually used. The symmetrization step of their Appendix A.6, which reduces the problem to a Gaussian process, requires only independence and unbiasedness. Moreover, the chaining argument that follows in their Lemmas A.17, A.21, and A.22 is carried out pointwise in the realization $S_G$. Therefore, the randomness of the sampler enters only at two further points. First, their event $\xi_G$ in their Eq. (10) asserts that the per-cluster weight sums concentrate. Under Bernoulli sampling, the summands are independent, and by the hatted Observation A.4 they are bounded by $\max_a 1/q_a \leq 2^{O(p)} k|P_i \cap G|/\Gamma_G$, so Bernstein's inequality yields the same failure probability $k \cdot \exp(-\Omega(\epsilon^2 \Gamma_G/k))$ as their Lemma A.13. Second, their Eq. (14) and the end of their Lemma A.22 use the deterministic sample-size bound $|P_i \cap S_G| \leq \Gamma_G$, and we replace it by the event $|S_G| \leq 4\Gamma_G$. Since $\mathbb{E}[|S_G|] = \sum_{a \in G} q_a \leq 2\Gamma_G$, by the Chernoff bound this event fails with probability at most $\exp(-\Omega(\Gamma_G))$. On this event, the same chains hold with constants doubled, and the covering numbers are evaluated at $4\Gamma_G$ instead of $\Gamma_G$. This costs only a constant factor. Their Definitions A.8, A.10 and A.18 impose their conditions point by point, through sets and

errors that depend only on $G$, $C$ and the point, so coverings of the parts of a partition of a set $S \subseteq G$ combine into a covering of $S$ at the same $\alpha$. Their covering number $\mathcal{N}_G(\Gamma, \alpha)$ is a maximum over the sets $S \subseteq G$ with $|S| \le \Gamma$. Hence, splitting a set of size at most $m\Gamma$ into $m$ parts of size at most $\Gamma$ gives $\mathcal{N}_G(m\Gamma, \alpha) \le \mathcal{N}_G(\Gamma, \alpha)^m$ for every integer $m \ge 1$, and likewise for the outer and layered covering numbers. With $m = 4$, each entropy integral at most doubles, while the factor $1/\sqrt{\Gamma_G}$ in their chaining bound comes from the budget and is unchanged, so the error bound grows by a constant factor. The same inequality shows that their Eqs. (8) and (9) still hold after $\Gamma_G$ is multiplied by an integer. For the budgets of Step 4 nothing is lost, since the bounds of their Lemma B.2 do not depend on $\Gamma_G$. The complement is handled in the same way as they handle $\bar{\xi}_G$ in their Eqs. (14)–(15). The crude deterministic bound grows by the factor $|S_G|/\Gamma_G$, and the expectation of this factor restricted to the complement is $\exp(-\Omega(\Gamma_G)) \le \epsilon/(4k)$ because $\Gamma_G = \Omega(k\epsilon^{-2}\log(k\epsilon^{-1}))$.

*Step 3: center weights.* Item 3 of their Lemma A.12 is linear in the center weights, so replacing $|P_i \setminus \mathcal{G}|$ by $\widehat{w}(a_i^\star)$ changes its left-hand side by at most $\sum_{i\in[k]} \epsilon|P_i| \cdot \mathrm{dist}^p(a_i^\star, C)$. By averaging the relaxed triangle inequality over $P_i$, we have $|P_i| \cdot \mathrm{dist}^p(a_i^\star, C) \le 2^p \cdot (\mathrm{cost}(P_i, C) + \mathrm{cost}(P_i, A^\star))$. Therefore, the total change is at most $2^p\epsilon \cdot \mathrm{cost}(A, C + A^\star)$, and their Item 3 holds with $\epsilon$ replaced by $(1 + 2^p)\epsilon$.

We sum the three items over the $O_p(\log^2(k/\epsilon))$ groups and apply Markov's inequality at threshold $500\epsilon$ to the expectation bound $4\epsilon$ of their proof of Theorem A.11, which gives failure probability $0.008$. Note that they use the threshold $40\epsilon$ and probability $0.9$ instead. The remaining $0.002$ of the allowed failure probability $0.01$ is charged to the events $|S_G| \le 4\Gamma_G$ of Step 2, which the size bound uses. Over the $O_p(\log^2(k/\epsilon))$ groups, the failure probabilities of these events sum to at most $\sum_G \exp(-\Omega(\Gamma_G)) \le 0.002$, once the constant factor in $\Gamma_G = \Omega(k\epsilon^{-2}\log(k\epsilon^{-1}))$ is large enough. After rescaling $\epsilon$ by $2^{O(p)}$, we obtain the first claim. Their proof bounds the expected error by $4\epsilon \cdot \mathrm{cost}(P, C + A^\star) = 4\epsilon\,(\mathrm{cost}(P, C) + \mathrm{cost}(P, A^\star))$, and hence the error is at most $40\epsilon \cdot \mathrm{cost}(P, C + A^\star)$ with probability $0.9$, and at most $500\epsilon \cdot \mathrm{cost}(P, C + A^\star)$ except with probability $0.008$, for all $C$ simultaneously. This is because their Lemma A.12 normalizes every group by $\mathrm{cost}(G, C + A^\star)$ or $\mathrm{cost}(P^G, C + A^\star)$ and its third item by $\mathrm{cost}(P, C + A^\star)$. Moreover, their proof invokes the approximation guarantee of $A^\star$ only in its last line, to write $\mathrm{cost}(P, C + A^\star) \le O(1)\,\mathrm{cost}(P, C)$. The ring, group, and budget definitions and the statements of the three items use $A^\star$ only through the induced clusters, the averages $\Delta_i$, the costs $\mathrm{cost}(\cdot, A^\star)$, and the covering numbers. The bounds on these covering numbers in their Lemma B.2, which Step 4 uses, are uniform in $A^\star$. Items 2 and 3 are stated there in this normalization, with proofs cited from Lemma 13 of Cohen-Addad et al. [2022a] and Lemma 4 of Cohen-Addad et al. [2021], which are their references [14] and [18]. This gives the relative form of the last claim for an arbitrary $A^\star$. The size bound of the last claim also holds on the same event, since the events $|S_G| \le 4\Gamma_G$ of Step 2 do not use the approximation guarantee of $A^\star$.

*Step 4: sample budgets without their Assumption 1.* The budgets of their Theorem B.1 are proved under their Assumption 1, which asks for at most $\mathrm{poly}(k/\epsilon)$ distinct points, dimension $O(\epsilon^{-2}\log\|P\|_0)$, and unweighted input. The third condition holds for our input. The first is used only to turn $\log\|P\|_0$ into $O(\log(k/\epsilon))$ (it is also cited in the proof of their Eq. (35), a step of their proof of Lemma B.2, but only for this conversion). The second is used only where the $d$-branch of their Lemma B.2 bounds the small-$\alpha$ part of an entropy integral, in their Eqs. (26)–(27) for the main-group budget of their Eq. (8) and likewise for the outer-group budget of their Eq. (9). Their Lemma B.2 states both branches with no condition on $d$, and we apply it with $\log\|P\|_0 \le \log n$. This also covers the sets $A^\star \cup S$ that their proof embeds, which have at most $k + n \le 2n$ points, since $\log(2n) \le 2\log n$ for $n \ge 2$. The main-group $d$-branch of their Lemma B.2 comes from their Lemma B.7, which assumes $d \ge \log k$. We may assume this by appending zero coordinates up to dimension $d' := \max\{d, \lceil\log k\rceil\}$. This changes no distance, and a covering for all center sets in $(\mathbb{R}^{d'})^k$ is also one for all center sets in $(\mathbb{R}^d)^k$, so we apply their Lemma B.2 in $\mathbb{R}^{d'}$. The padding is used only in the analysis, and the algorithm is unchanged. Let $L := \log(2d'k/\epsilon) \le 2\log(2dk/\epsilon)$ and $\alpha_0 := \epsilon/\sqrt{d'}$. We split each integral at $\alpha_0$ instead of $\epsilon$, using the $d$-branch only below $\alpha_0$. For main groups, the part below $\alpha_0$ is $O_p(\sqrt{kd'} \cdot \alpha_0\sqrt{L}) = O_p(\epsilon\sqrt{kL})$. Above $\alpha_0$, their Eq. (28), started at $\alpha_0$ and with $\log n$ in place of $\log(k\epsilon^{-1})$, together with $\log\epsilon^{-1} \le L$, $\log(\epsilon^{-1}\alpha^{-1}) \le 2L$ and $\int_{\alpha_0}^{(8p)^p\epsilon^{-p+1}} \alpha^{-1}d\alpha = O_p(L)$, gives $O_p(\sqrt{k^{\frac{2p+2}{p+2}}L^4\log n})$. Their Lemma B.2 is stated only for $0 < \alpha \le 1$, while their Eq. (28) uses

it up to $(8p)^p\epsilon^{-p+1}$. For $p \le 2$ we need it only on $[\alpha_0, 1]$. Indeed, for $a \in P_i$ let $c = \operatorname{dist}(a, C)$ and $r = \operatorname{dist}(a, A^\star) = \operatorname{dist}(a, a_i^\star)$. Since $|\operatorname{dist}(a, C) - \operatorname{dist}(a_i^\star, C)| \le r$, the mean value theorem gives $|\operatorname{dist}^p(a, C) - \operatorname{dist}^p(a_i^\star, C)| \le pr(c+r)^{p-1} \le p2^{p-1}r\max\{c, r\}^{p-1} \le p2^{p-1}(\sqrt{c^p r^p} + r^p)$, where the last step uses $(c/r)^{p/2-1} \le 1$ for $c \ge r$ and $p \le 2$. Their relative covering error in Definition A.8 is $\sqrt{c^p r^p} + r^p$ times $\sqrt{(\operatorname{cost}(G, C) + \operatorname{cost}(G, A^\star))/\operatorname{cost}(G, A^\star)} \ge 1$, so the zero vector is an $\alpha$-covering for every $\alpha \ge p2^{p-1}$. Covering numbers are nonincreasing in $\alpha$, so on $[1, p2^{p-1}]$ the integrand is at most its value at $\alpha = 1$, and this piece contributes $O_p(\sqrt{k^{\frac{2p+2}{p+2}} L^2 \log n})$. For $p > 2$ we follow their Eq. (28), which applies the bound of their Lemma B.2 on $(1, (8p)^p\epsilon^{-p+1}]$, beyond its stated range. For $p > 2$ our main-group budget therefore rests on this step of their proof of Theorem B.1. For outer groups, the covering number is 1 for $\alpha \ge 1$. Indeed, their Definition A.10 asks for $|\operatorname{dist}^p(a, C) - v_a| \le \alpha(\operatorname{dist}^p(a, C) + \operatorname{dist}^p(a, A^\star))$ point by point, and the zero vector satisfies this for every $\alpha \ge 1$. Below $\alpha_0$, the $d$-branch of their outer bound gives $\int_0^{\alpha_0} O_p(\sqrt{kd' \log \alpha^{-1}})\, d\alpha = O_p(\sqrt{kd'} \cdot \alpha_0\sqrt{L}) = O_p(\epsilon\sqrt{kL})$. On $[\alpha_0, 1]$, its other branch gives $\int_{\alpha_0}^1 O_p(\alpha^{-1}\sqrt{k \log n \log \alpha^{-1}})\, d\alpha = O_p(\sqrt{kL \log n} \cdot \log \alpha_0^{-1}) = O_p(\sqrt{kL^3 \log n})$, since $\log \alpha_0^{-1} \le L$. Their outer bound cites Lemma 21 of Cohen-Addad et al. [2022a]. In the full version (arXiv:2202.12793), the $\alpha^{-2}$-branch is Lemma 21 and the $d$-branch is Lemma 22. Both lemmas are stated only for integer $p$ and $\alpha \le 1/2$, with $\log(1/(\alpha\epsilon))$ and $\log(4p/(\alpha\epsilon))$ respectively in place of $\log(1/\alpha)$. This changes the two parts only by factors $O_p(1)$, since $\log(1/(\alpha\epsilon)) \le 2L$ for $\alpha \ge \alpha_0$ and $\log(4p/(\alpha\epsilon)) \le \log(4p) + 2\log \alpha^{-1}$ for $\alpha \le \alpha_0 \le \epsilon$. The proof of Lemma 22 places a net around every input point and chooses $k$ net points, so its bound carries an additive $k \log \|P\|_0 \le k \log n$ that the statement omits. Below $\alpha_0$ this adds $O(\alpha_0\sqrt{k \log n}) \le O(\epsilon\sqrt{k \log n})$. The covering number is nonincreasing in $\alpha$, so the range $(1/2, 1]$ adds at most $O_p(\sqrt{kL \log n})$. The outer budget is unchanged. For non-integer $p$ we use the outer bound of their Lemma B.2 as stated, as their proof of Theorem B.1 does. These bounds do not depend on $\Gamma_G$, so their self-referential Eqs. (8) and (9) hold for $\Gamma_G = O_p(k^{\frac{2p+2}{p+2}}\epsilon^{-2}L^4 \log n)$ and $\Gamma_G = O_p(k\epsilon^{-2}L^3 \log n)$, respectively, since both dominate their term $k\epsilon^{-2}\log(k/\epsilon)$. Compared with their Theorem B.1, the main-group budget replaces $\log(k/\epsilon)$ by $\log n$ and $\log^4 \epsilon^{-1}$ by $L^4$. Summing over the $O_p(\log^2(k/\epsilon))$ groups gives the stated coreset size. □

We obtain a discrete constant-factor solution without constructing a coreset, using the bicriteria centers of Lemma H.2, the cell counts of Lemma H.5, and the following two facts.

**Lemma H.11** (Theorem 1 of Dürr and Høyer [1996])**.** *Given query access to $N$ values, there is a quantum algorithm that outputs an index of a minimum value with probability at least $1/2$ using $O(\sqrt{N})$ queries, and repeating it $O(\log(1/\delta))$ times and returning the smallest value found succeeds with probability at least $1 - \delta$. In particular, for $c \in \mathbb{R}^d$ a nearest point of $A$ to $c$ can be found with probability at least $1 - \delta$ using $\widetilde{O}(\sqrt{n}\log(1/\delta))$ queries to the points of $A$ and $\widetilde{O}(\sqrt{n}d\log(1/\delta))$ time.*

**Lemma H.12** (Cell counts as a proxy for the clustering cost)**.** *Let $x' = (x'_1, \ldots, x'_m) \in (\mathbb{R}^d)^m$, let $\gamma \ge 1$, let $\tau : A \to x'$ satisfy $\|a - \tau(a)\|_2^p \le \gamma \cdot \operatorname{cost}(a, x')$ for all $a \in A$, let $C_j = \{a \in A : \tau(a) = x'_j\}$ and $n_j = |C_j|$, and define, for any finite $C \subset \mathbb{R}^d$, the weighted cost $W(C) = \sum_{j=1}^m n_j \cdot \operatorname{dist}^p(x'_j, C)$. Then for every finite $C \subset \mathbb{R}^d$,*

$$W(C) \le 2^{p-1}\big(\gamma \cdot \operatorname{cost}(A, x') + \operatorname{cost}(A, C)\big) \qquad \text{and}$$
$$\operatorname{cost}(A, C) \le 2^{p-1}\big(\gamma \cdot \operatorname{cost}(A, x') + W(C)\big).$$

*Proof.* Fix $a \in C_j$. By the triangle inequality and the relaxed triangle inequality $(u+v)^p \le 2^{p-1}(u^p + v^p)$ for $u, v \ge 0$ and $p \ge 1$, we have

$$\begin{aligned}\operatorname{dist}^p(x'_j, C) &\le (\|x'_j - a\|_2 + \operatorname{dist}(a, C))^p \le 2^{p-1}(\|a - \tau(a)\|_2^p + \operatorname{dist}^p(a, C)) \\ &\le 2^{p-1}(\gamma \cdot \operatorname{cost}(a, x') + \operatorname{dist}^p(a, C)).\end{aligned}$$

Since the cells partition $A$, summing over $a \in C_j$ and then over $j \in [m]$ gives $W(C) = \sum_j \sum_{a \in C_j} \operatorname{dist}^p(x'_j, C) \le 2^{p-1}(\gamma \cdot \operatorname{cost}(A, x') + \operatorname{cost}(A, C))$. Symmetrically, for $a \in C_j$ we have $\operatorname{dist}^p(a, C) \le (\|a - x'_j\|_2 + \operatorname{dist}(x'_j, C))^p \le 2^{p-1}(\gamma \cdot \operatorname{cost}(a, x') + \operatorname{dist}^p(x'_j, C))$, and summing gives the second inequality. □

**Lemma H.13** (Discrete constant-factor solution in $\widetilde{O}(n^{0.5}kd + d \cdot \mathrm{poly}(k))$ time)**.** *Let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$, $k \geq 1$, $p \geq 1$ a constant and $\delta \in (0, 1/2)$. There is a quantum algorithm that, with probability at least $1 - \delta$, outputs a set $A^\star \subseteq A$ of at most $k$ input points with $\mathrm{cost}(A, A^\star) \leq 2^{O(p)} \cdot \min_{x \in (\mathbb{R}^d)^k} \mathrm{cost}(A, x)$, using $\widetilde{O}(n^{0.5}k \log(1/\delta))$ queries to the points of $A$, $\widetilde{O}(n^{0.5}kd \log(1/\delta)) + \widetilde{O}(d \cdot \mathrm{poly}(k))$ time, and $\mathrm{poly}(k \log(n/\delta))$ additional preprocessing time.*

*Proof.* Let $\mathrm{OPT} = \min_{x \in (\mathbb{R}^d)^k} \mathrm{cost}(A, x)$. We run each step below with failure probability at most $\delta/4$.

1. Generate $x' \in (\mathbb{R}^d)^m$, $m = O(k \log^2 n)$, with $\mathrm{cost}(A, x') \leq 2^{O(p)} \cdot \mathrm{OPT}$ by Lemma H.2. We amplify the success probability from $5/6$ to $1 - \delta/4$ by $O(\log(1/\delta))$ independent runs, and keep the candidate of least estimated cost, using Lemma H.4 at constant accuracy with a constant loss absorbed into $2^{O(p)}$. By the size cap of Lemma H.2, every candidate has at most $c_{\mathrm{bi}} k \lceil \log_2 n \rceil^2$ centers on every outcome, so the selected one does too, even if it comes from a failed run. Each $\mathrm{cost}(a_i, x')$ is a minimum over $m = \widetilde{O}(k)$ centers, so each candidate takes $\widetilde{O}(\sqrt{n}kd)$ time, and hence the step takes $\widetilde{O}(\sqrt{n}kd \log(1/\delta))$ time and $\widetilde{O}(\sqrt{n}k \log(1/\delta))$ queries in total.

2. Generate the oracle $\tau$ of Lemma H.3 on $A, x'$ with $c_\tau \in [2.5, 3)$, and also generate estimates $\widetilde{n}_j \in [\frac{3}{4}n_j, \frac{5}{4}n_j]$ of the cell sizes $n_j = |\{a \in A : \tau(a) = x'_j\}|$ by Lemma H.5 with accuracy $1/4$. This step takes $\widetilde{O}(\sqrt{nm}d) = \widetilde{O}(\sqrt{nk}d)$ time and $\widetilde{O}(\sqrt{nk})$ queries.

3. Consider the weighted instance $\widetilde{W}(C) = \sum_{j=1}^m \widetilde{n}_j \cdot \mathrm{dist}^p(x'_j, C)$ on the $m$ weighted points $(x'_j, \widetilde{n}_j)$. Compute $\widetilde{C} \in (\mathbb{R}^d)^k$ with $\widetilde{W}(\widetilde{C}) \leq 2^{O(p)} \cdot \min_{C \in (\mathbb{R}^d)^k} \widetilde{W}(C)$ by applying Lemma H.17 to the weighted set, in $d \cdot \mathrm{poly}(k)$ time and no queries to $A$. Note that Lemma H.17 is stated for unweighted inputs, but the weights $\widetilde{n}_j$ here are polynomially bounded, and any polynomial-time constant-factor approximation for weighted $(k, p)$-clustering suffices.

4. For each $c \in \widetilde{C}$, find a nearest point $a(c) \in A$ by Lemma H.11, in $\widetilde{O}(\sqrt{n}d)$ time per center, and output $A^\star = \{a(c) : c \in \widetilde{C}\}$.

Correctness. Let $W$ be as in Lemma H.12 for this $x'$ and $\tau$, with $\gamma := c_\tau^p < 3^p$ by Lemma H.3. Then $\widetilde{W}(C) \in [\frac{3}{4}W(C), \frac{5}{4}W(C)]$ for every $C$. Let $C^\star$ attain OPT. By Step 3 and Lemma H.12, we have

$$\widetilde{W}(\widetilde{C}) \leq 2^{O(p)}\widetilde{W}(C^\star) \leq 2^{O(p)}W(C^\star) \leq 2^{O(p)} \cdot 2^{p-1}(\gamma \cdot \mathrm{cost}(A, x') + \mathrm{OPT}) \leq 2^{O(p)} \cdot \mathrm{OPT},$$

where we use $\mathrm{cost}(A, x') \leq 2^{O(p)}\,\mathrm{OPT}$ and $\gamma < 3^p$. The second inequality of Lemma H.12 then gives $\mathrm{cost}(A, \widetilde{C}) \leq 2^{p-1}(\gamma \cdot \mathrm{cost}(A, x') + \frac{4}{3}\widetilde{W}(\widetilde{C})) \leq 2^{O(p)} \cdot \mathrm{OPT}$. Finally, for every $a \in A$ and $c \in \widetilde{C}$, we have $\|a - a(c)\|_2 \leq \|a - c\|_2 + \|c - a(c)\|_2 \leq 2\|a - c\|_2$, since $a(c)$ is a nearest input point to $c$. Hence, $\mathrm{dist}(a, A^\star) \leq 2\,\mathrm{dist}(a, \widetilde{C})$ and $\mathrm{cost}(A, A^\star) \leq 2^p \cdot \mathrm{cost}(A, \widetilde{C}) \leq 2^{O(p)} \cdot \mathrm{OPT}$. A union bound over the four steps gives the failure probability, and summing the costs gives the time and query bounds, where the preprocessing of Lemma H.3 and of Lemma H.2 is $\mathrm{poly}(k \log(n/\delta))$. □

**Lemma H.14** (Center replacement for near-center points)**.** *Let $p \geq 1$, $\epsilon \in (0, 1]$, $a, a^\star \in \mathbb{R}^d$ and let $C \subset \mathbb{R}^d$ be finite. Then*

$$|\,\mathrm{dist}^p(a, C) - \mathrm{dist}^p(a^\star, C)| \leq \epsilon \cdot \mathrm{dist}^p(a, C) + (3p/\epsilon)^p \cdot \|a - a^\star\|_2^p.$$

*Proof.* Let $x := \mathrm{dist}(a, C)$, $y := \mathrm{dist}(a^\star, C)$, and $\eta := \|a - a^\star\|_2$. By the triangle inequality, we have $|x - y| \leq \eta$, so the mean value theorem gives $|x^p - y^p| \leq p\,|x - y| \max\{x, y\}^{p-1} \leq p\,\eta\,(x + \eta)^{p-1}$. If $\eta \leq \epsilon x/(2p)$, then $(x + \eta)^{p-1} \leq (1 + \epsilon/(2p))^{p-1}x^{p-1} \leq e^{1/2}x^{p-1}$, and therefore $|x^p - y^p| \leq p \cdot \frac{\epsilon x}{2p} \cdot e^{1/2}x^{p-1} \leq \epsilon x^p$. Otherwise, we have $x < 2p\eta/\epsilon$ and $y \leq x + \eta < (2p/\epsilon + 1)\eta \leq 3p\eta/\epsilon$, so $|x^p - y^p| \leq \max\{x, y\}^p \leq (3p/\epsilon)^p\eta^p$. □

**Theorem H.15** (Quantum coreset for $(k,p)$-clustering via the construction of Huang et al. [2024])**.** *Let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$, let $1 \le k \le n$ be an integer, $\epsilon \in (0,1)$, and let $p \ge 1$ be a constant. There exists a quantum algorithm that, with probability at least $0.9$, outputs an $\epsilon$-coreset of $A$ for $(k,p)$-clustering that consists of input points only and has size $k + O_p(k^{\frac{2p+2}{p+2}}\epsilon^{-2}\log^2(k/\epsilon)\log n \cdot \log^4(2dk/\epsilon)) = \widetilde{O}_p(k^{\frac{2p+2}{p+2}}\epsilon^{-2})$, in time*

$$\widetilde{O}_p(\epsilon^{-1}n^{0.5}d\,k^{\frac{2p+3}{p+2}}) + \widetilde{O}(d \cdot \mathrm{poly}(k)),$$

*using $\widetilde{O}_p(n^{0.5}(k+k^{\frac{p+1}{p+2}}\epsilon^{-1}))$ queries to the points of $A$. The size carries the factor $\log n \cdot \log^4(2dk/\epsilon)$, unlike Theorem B.1 of Huang et al. [2024], whose size depends on neither $n$ nor $d$.*

*Proof.* Throughout the proof, we run each randomized step with failure probability at most $0.01$, and we rescale $\epsilon$ by the constant $2^{O(p)}$ required by Lemma H.10. Every stage runs on the input points in $\mathbb{R}^d$ without dimension reduction, and the stages below charge $O(kd)$ time for each access to a point after Stage 1. No assumption on $d$ is needed, since the budgets of Lemma H.10 hold for every $d$.

*Stage 1: a constant-factor discrete solution.* Run Lemma H.13 to obtain $A^\star \subseteq A$ with at most $k$ points and $\mathrm{cost}(A, A^\star) \le 2^{O(p)} \cdot \min_{x \in (\mathbb{R}^d)^k} \mathrm{cost}(A,x)$, in $\widetilde{O}(n^{0.5}kd) + d \cdot \mathrm{poly}(k)$ time and $\widetilde{O}(n^{0.5}k)$ queries.

*Stage 2: decomposition and statistics.* Given $A^\star$, the cluster $P_i$ of a point and its value $\mathrm{dist}^p(a, A^\star)$ can be computed in $O(kd)$ time. We obtain all cell statistics of this stage from Corollary H.8 which estimates all cells of a partition at once but requires values of bounded range, and we ensure the bounded range by truncation.

*(a) Truncation.* We estimate $\Lambda := \mathrm{cost}(A, A^\star)$ within a factor 2 by Lemma H.4, which gives $\widehat{\Lambda} \in [\Lambda/2, 2\Lambda]$ in $\widetilde{O}(n^{0.5})$ queries and $\widetilde{O}(n^{0.5}kd)$ time, and we then set the floor $F := \epsilon^{p+1}\widehat{\Lambda}/(2(3p)^p n) \le \epsilon^{p+1}\Lambda/((3p)^p n)$. Let $T := \{a \in A : \mathrm{dist}^p(a, A^\star) \le F\}$ be the truncated points, $T_i := T \cap P_i$, $A' := A \setminus T$, and $P_i' := P_i \setminus T$. If $\widehat{\Lambda} = 0$, then $\Lambda = 0$ and $T = A$, and hence $A^\star$ with the estimated cluster masses of Lemma H.5 is an $\epsilon$-coreset by Lemma H.14 with $\|a - a^\star\|_2 = 0$. Thus, we assume $\widehat{\Lambda} > 0$ below. Every $a \in A'$ has $\mathrm{dist}^p(a, A^\star) \in (F, \Lambda] \subseteq [F, RF]$ with $R := 4(3p)^p n/\epsilon^{p+1}$, and therefore $\log R = O_p(\log(n/\epsilon))$. The oracle that maps $a \in T$ to a null cell and reports the value $0$ for it can be computed in $O(kd)$ time, and the points of $T$ will be represented by their centers.

*(b) Cluster statistics.* We apply Corollary H.8 with the $k$ clusters $P_i'$ as cells, in $\widetilde{O}_p(\sqrt{nk} \cdot kd) = \widetilde{O}_p(n^{0.5}k^{1.5}d)$ time. This estimates all $|P_i'|$ and $\mathrm{cost}(P_i', A^\star)$ within $1 \pm 1/4$, and since $(5/4)/(3/4) < 2$, we obtain estimates $\widehat{\Delta}_i \in [\Delta_i'/2, 2\Delta_i']$ of the averages $\Delta_i' := \mathrm{cost}(P_i', A^\star)/|P_i'|$. If $P_i'$ is empty, then the cluster has no rings and $a_i^\star$ receives the weight $|T_i|$ in (d). The estimates $\widehat{\Delta}_i$ fix the rings of Huang et al. [2024] on the clusters $P_i'$, as in their Definition A.2, and the ring index of a point can be computed in $O(kd)$ time. Their definition has $O_p(\log(k/\epsilon))$ main rings per cluster plus inner and outer rings, and there are at most $O(\log R)$ outer rings per cluster since every value lies in $[F, \Lambda]$. Thus, there are $M = \widetilde{O}_p(k)$ nonempty rings in total.

*(c) Ring statistics.* Applying Corollary H.8 again with the $M$ nonempty rings as cells, we obtain all ring masses and ring costs within $1 \pm 1/4$ in $\widetilde{O}_p(\sqrt{nM} \cdot kd) = \widetilde{O}_p(n^{0.5}k^{1.5}d)$ time. This fixes the groups and the group-cost estimates $\widehat{c}_G$, which are sums of ring costs and are within $1 \pm 1/4$.

*(d) Center weights.* Each center $a_i^\star$ must receive the weight $|T_i| + |P_i' \setminus \mathcal{G}|$ within additive error $\epsilon|P_i|$. To estimate these weights, we run Lemma H.5 with relative accuracy $\epsilon/2^{O(p)}$ on the cells $\{T_i\}_i \cup \{\text{rings}\}$, in $\widetilde{O}_p(\sqrt{nM/\epsilon} \cdot kd) = \widetilde{O}_p(\epsilon^{-0.5}n^{0.5}k^{1.5}d)$ time.

*Stage 3: one-shot sampling.* Sample every point $a \in A'$ independently with probability $q_a = \min\{1, \Gamma_G \cdot \mathrm{dist}^p(a, A^\star)/\widehat{c}_G\}$, where $G$ is the group of $a$'s ring, and $q_a = 0$ for points of $T$ and for points outside all groups, with the budgets $\Gamma_G$ that Lemma H.10 gives for $A'$ and the integer $\max\{2,n\} \ge \max\{2, |A'|\}$, which is also at least $k$ by hypothesis. Each probability is computable in $O(kd)$ time after Stage 2, and $\sum_a q_a \le 2\sum_G \Gamma_G = \widetilde{O}_p(k^{\frac{2p+2}{p+2}}\epsilon^{-2}) =: s$ over the $O_p(\log^2(k\epsilon^{-1}))$ groups, so Lemma B.15 performs the sampling in $\widetilde{O}(\sqrt{ns} \cdot kd) = \widetilde{O}_p(\epsilon^{-1}n^{0.5}k^{\frac{2p+3}{p+2}}d)$ time.

*Correctness.* Let $S$ be the set of sampled points with weights $1/q_a$, and let each center $a_i^\star$ receive the estimated weight $\widehat{w}_i$ of $|T_i| + |P_i' \setminus \mathcal{G}|$. We fix $C \in (\mathbb{R}^d)^k$ and split the error into three parts.

First, the sampler runs on $A'$ with the center set $A^\star$ and the clusters $P_i'$, which are built from factor-2 estimates. Its Bernoulli weights are the reciprocals of the probabilities used, and it uses the exact center weights $|P_i' \setminus \mathcal{G}|$. The estimation error of these weights is $\epsilon|P_i|$ rather than the $\epsilon|P_i'|$ the lemma tolerates, and we charge it separately in the third part. Since $A^\star$ need not be approximately optimal for $A'$, we use the relative form of Lemma H.10. With probability at least $0.99$, we have for all $C$,

$$\Big|\sum_{s\in S} w_s \operatorname{dist}^p(s,C) + \sum_i |P_i' \setminus \mathcal{G}| \cdot \operatorname{dist}^p(a_i^\star, C) - \operatorname{cost}(A', C)\Big| \\ \le O(\epsilon)\left(\operatorname{cost}(A', C) + \operatorname{cost}(A', A^\star)\right) \le O(\epsilon)\left(\operatorname{cost}(A, C) + \Lambda\right).$$

Second, we represent the truncated points by their centers. Applying Lemma H.14 to each $a \in T_i$ with $a^\star = a_i^\star$ and $\|a - a_i^\star\|_2^p = \operatorname{dist}^p(a, A^\star) \le F$, we have

$$\Big|\sum_i |T_i| \cdot \operatorname{dist}^p(a_i^\star, C) - \operatorname{cost}(T, C)\Big| \le \sum_{a\in T} \left(\epsilon \operatorname{dist}^p(a, C) + (3p/\epsilon)^p \operatorname{dist}^p(a, A^\star)\right) \\ \le \epsilon \operatorname{cost}(A, C) + (3p/\epsilon)^p nF \le \epsilon \operatorname{cost}(A, C) + \epsilon\Lambda.$$

Third, the center weights are estimated within additive error $\epsilon|P_i|$, which by the relaxed triangle inequality averaged over $P_i$ as in Step 3 of the proof of Lemma H.10 changes the coreset cost by at most $\sum_i \epsilon|P_i| \operatorname{dist}^p(a_i^\star, C) \le 2^p\epsilon\left(\operatorname{cost}(A, C) + \Lambda\right)$. Summing the three parts, the weighted set $S \cup A^\star$ has cost within $O_p(\epsilon)\left(\operatorname{cost}(A, C) + \Lambda\right)$ of $\operatorname{cost}(A, C) = \operatorname{cost}(A', C) + \operatorname{cost}(T, C)$ for all $C$. Moreover, by Stage 1, $\Lambda = \operatorname{cost}(A, A^\star) \le 2^{O(p)} \min_x \operatorname{cost}(A, x) \le 2^{O(p)} \operatorname{cost}(A, C)$. Rescaling $\epsilon$ by $2^{O(p)}$ gives an $\epsilon$-coreset. The at most seven randomized steps, each failing with probability at most $0.01$, are Stage 1, the estimate $\widehat{\Lambda}$, the two applications of Corollary H.8, the counts of (d), the guarantee of Lemma H.10 and the sampler of Lemma B.15, so by a union bound, the total failure probability is at most $0.07 < 0.1$. The size of the coreset is $k + 2^{O(p)} \sum_G \Gamma_G = k + O_p(k^{\frac{2p+2}{p+2}} \epsilon^{-2} \log^2(k/\epsilon) \log n \cdot \log^4(2dk/\epsilon))$, since the truncation only removes points from the sampled set and the budgets of Lemma H.10 hold for every $d$ without any preprocessing of the input. Here the factor $\log^2(k/\epsilon)$ accounts for the $O_p(\log^2(k/\epsilon))$ groups, and $\log n \cdot \log^4(2dk/\epsilon)$ is the factor of the main-group budgets, which dominate the outer-group budgets. Theorem B.1 of Huang et al. [2024] has $\log(k/\epsilon) \log^4 \epsilon^{-1}$ in place of this factor. Finally, every element of the coreset is an input point.

*Running time and queries.* Stage 1 costs $\widetilde{O}(n^{0.5}kd) + \widetilde{O}(d \cdot \operatorname{poly}(k))$ time and $\widetilde{O}(n^{0.5}k)$ queries, Stage 2 costs $\widetilde{O}_p(n^{0.5}k^{1.5}d + \epsilon^{-0.5}n^{0.5}k^{1.5}d)$ time and $\widetilde{O}_p(\epsilon^{-0.5}n^{0.5}k^{0.5})$ queries, and Stage 3 costs $\widetilde{O}_p(\epsilon^{-1}n^{0.5}k^{\frac{2p+3}{p+2}}d)$ time and $\widetilde{O}_p(\epsilon^{-1}n^{0.5}k^{\frac{p+1}{p+2}})$ queries. Since $k \le k^{1.5} \le k^{\frac{2p+3}{p+2}}$ for $p \ge 1$, the total time is $\widetilde{O}_p(\epsilon^{-1}n^{0.5}d\,k^{\frac{2p+3}{p+2}}) + \widetilde{O}(d \cdot \operatorname{poly}(k))$ and the total number of queries is $\widetilde{O}_p(n^{0.5}(k + \epsilon^{-1}k^{\frac{p+1}{p+2}}))$. □

### H.2 Quantum Algorithm for Data Selection

As an application, we study the *data selection pipeline* in machine learning, where the goal is to select a weighted subset of data points for training or fine-tuning the model while preserving desirable properties. In this model, data are given as $d$-dimensional embeddings, and a loss function $\ell : \mathbb{R}^d \to \mathbb{R}_{\ge 0}$ is used to grade the quality of the embedding. $\ell$ can be expensive to evaluate, such as a deep neural network. Axiotis et al. [2024] provides a principled way for data selection using the coreset of $(k, p)$-clustering, under some mild assumptions on $\ell$.

**Assumption H.16.** *Let $\Lambda = (\Lambda_1, \ldots, \Lambda_k) \in \mathbb{R}^k_{\ge 0}, x \in (\mathbb{R}^d)^k$ and let $E \subseteq \mathbb{R}^d$ be a set of embeddings, we say the loss function $\ell$ is $(p, \Lambda)$-well-behaved with respect to $E$ and $x$ if, for every $x_j \in x$ and every $e \in C_j$, where $C_j = \{e \in E : \arg\min_{x_i \in x} \|x_i - e\|_2^p = x_j\}$,*

$$|\ell(e) - \ell(x_j)| \le \Lambda_j \|e - x_j\|_2^p.$$

Define the weighted cost as $\operatorname{cost}^\Lambda(a_i, x) = \Lambda_{j(i)} \operatorname{cost}(a_i, x)$ where we use $j(i)$ to denote the index of the cluster that $a_i$ is assigned to, and similarly $\operatorname{cost}^\Lambda(A, x) = \sum_{i=1}^n \operatorname{cost}^\Lambda(a_i, x) =$

$\sum_{i=1}^{k} \Lambda_i \sum_{a_j \in C_i} \|a_j - x_i\|_2^p$. Axiotis et al. [2024] essentially proves that under Assumption H.16, one could perform weighted sampling according to $\mathrm{cost}^{\Lambda}(a_i, x)$. In addition, the expensive loss function only needs to be evaluated on the centers. For convenience, we state an approximate $k$-centers algorithm below.

**Lemma H.17** (Mettu and Plaxton [2004])**.** *Let $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^d$ and $X = (\mathbb{R}^d)^k$, let $\delta \in (0, 1)$. Then, there exists an algorithm that computes a solution $x' \in X$ such that*

$$\mathrm{cost}(A, x') \leq 2^{O(p)} \cdot \min_{x \in X} \mathrm{cost}(A, x),$$

*holds with probability at least $1 - \delta$. Moreover, $x'$ can be computed in time*

$$O(ndk + nd \log(n/\delta) + k^2 \log^2 n + \log^2(1/\delta) \log^2 n) = \widetilde{O}(ndk).$$

We now state a quantum implementation of the adaptive sampling due to Axiotis et al. [2024].

**Algorithm 13** Quantum one-round adaptive sampling for data selection.

**procedure** QDATASELECTION($A \in \mathbb{R}^{n \times d}, x \in (\mathbb{R}^d)^k, \Lambda \in \mathbb{R}^k_{\geq 0}, \ell : \mathbb{R}^d \to \mathbb{R}_{\geq 0}, \epsilon \in (0, 1)$)
  $s \leftarrow 300\epsilon^{-2}, \epsilon' \leftarrow 0.01, \delta \leftarrow 0.001$
  Let $\tau : A \to x$ be such that $\tau(a_i) = \arg\min_{x_j \in x} \|a_i - x_j\|_2^p$, and let $j(i)$ denote the index of $\tau(a_i)$ in $x$
  $M \leftarrow \max_{i \in [n]} \Lambda_{j(i)} \|a_i - \tau(a_i)\|_2^p$ via Lemma H.11 with failure probability $\delta$
  Generate $\widetilde{\mathrm{cost}}^{\Lambda}(A, x) = (1 \pm \epsilon') \mathrm{cost}^{\Lambda}(A, x)$ via Lemma H.4 with the bound $M$, accuracy $\epsilon'$ and failure probability $\delta$
  Let $C_j = \{a \in A : \tau(a) = x_j\}$ and $n_j = |C_j|$
  Generate $\widetilde{n}_1, \ldots, \widetilde{n}_k$ via Lemma H.5 using $\tau$ with accuracy $\epsilon'$ and failure probability $\delta$
  Compute $\ell(x_1), \ldots, \ell(x_k)$
  $\mathrm{sum} \leftarrow \sum_{j=1}^{k} \widetilde{n}_j \cdot \ell(x_j)$
  Implement an oracle for each $a_i \in A$ as follows:
    $\widehat{\ell}(a_i) \leftarrow \ell(\tau(a_i)), v(a_i) \leftarrow \Lambda_{j(i)} \cdot \|a_i - \tau(a_i)\|_2^p$
    $q_i \leftarrow \frac{\widehat{\ell}(a_i) + v(a_i)}{\widetilde{\mathrm{cost}}^{\Lambda}(A,x) + \mathrm{sum}}$ ▷ $q_i \leftarrow 0$ if the denominator is 0
    $p_i \leftarrow \min\{1, s \cdot q_i\}$
  $D' \leftarrow$ QSAMPLE$(p, 0.001)$, and $w_i \leftarrow 1/p_i$ for $i \in D'$, $w_i \leftarrow 0$ for $i \notin D'$
  **return** $(D', w)$
**end procedure**

We then prove Algorithm 13 implements the data selection procedure in sublinear time.

**Theorem H.18.** *Let $\epsilon \in (0, 1), p \geq 1, \Lambda \in \mathbb{R}^k_{\geq 0}$, $A \in (\mathbb{R}^d)^n$ and $\ell$ be a loss function that is $(p, \Lambda)$-well-behaved with respect to $A$ and a clustering $x \in (\mathbb{R}^d)^k$. Then, there exists a quantum algorithm (Algorithm 13) that outputs a weight vector $w \in \mathbb{R}^n_{\geq 0}$ such that, with probability at least 0.99, $\|w\|_0 = O(\epsilon^{-2})$ and*

$$|\sum_{i=1}^{n} \ell(a_i) - \sum_{i=1}^{n} w_i \ell(a_i)| \leq \epsilon \cdot (\sum_{i=1}^{n} \ell(a_i) + 2\,\mathrm{cost}^{\Lambda}(A, x)).$$

*Moreover, the algorithm makes at most $k$ queries to the loss function $\ell$, $\widetilde{O}(n^{0.5}(\epsilon^{-1} + k^{0.5}))$ queries to the points in $A$, and uses an additional $\widetilde{O}(n^{0.5} k d(\epsilon^{-1} + k^{0.5}))$ time.*

*Proof.* Let $u_i := \widehat{\ell}(a_i) + v(a_i)$ and $U := \sum_{i=1}^{n} u_i = \sum_{j=1}^{k} n_j \ell(x_j) + \mathrm{cost}^{\Lambda}(A, x)$. Since $\tau(a_i)$ is the nearest center of $a_i$, Assumption H.16 gives $\ell(a_i) \leq u_i$ and $\widehat{\ell}(a_i) \leq \ell(a_i) + v(a_i)$, and therefore $L := \sum_{i=1}^{n} \ell(a_i) \leq U \leq L + 2\,\mathrm{cost}^{\Lambda}(A, x)$. Suppose that Lemmas H.11, H.4, and H.5 succeed, where each is run with failure probability $\delta = 0.001$. Then the denominator $\widetilde{D} := \widetilde{\mathrm{cost}}^{\Lambda}(A, x) + \mathrm{sum}$ satisfies $\widetilde{D} = (1 \pm \epsilon')U$, and hence $q_i = u_i / \widetilde{D}$ and $\sum_i p_i \leq s \sum_i q_i \leq s/(1 - \epsilon') \leq 1.02s$.

Algorithm 13 is the one-round adaptive sampling of Theorem 2 of Axiotis et al. [2024], with the exact statistics replaced by these estimates and with independent inclusions, and we verify its guarantee directly. The estimator $Z := \sum_i w_i \ell(a_i) = \sum_i \xi_i \ell(a_i)/p_i$, where the $\xi_i \sim \mathrm{Ber}(p_i)$ are independent, satisfies $\mathbb{E}\, Z = L$, and an index with $u_i = 0$ has $\ell(a_i) = 0$ and contributes to neither side. Moreover,

$$\begin{aligned}\operatorname{Var} Z &= \sum_{i:\, p_i<1} \ell(a_i)^2\, \frac{1-p_i}{p_i} \le \sum_i \frac{\ell(a_i)^2}{s\, q_i} = \frac{\widetilde{D}}{s} \sum_i \frac{\ell(a_i)^2}{u_i} \\ &\le \frac{\widetilde{D}\, L}{s} \le \frac{1.01\,(L + 2\,\mathrm{cost}^{\Lambda}(A,x))^2}{s},\end{aligned}$$

where we use $\ell(a_i) \le u_i$ and $\widetilde{D} \le 1.01\, U$. Then Chebyshev's inequality with $s = 300\epsilon^{-2}$ gives $|Z - L| \le \epsilon(L + 2\,\mathrm{cost}^{\Lambda}(A,x))$ except with probability $1.01/300 < 0.0034$. The size $\|w\|_0 = |D'|$ is a sum of independent Bernoulli variables with mean at most $1.02s$, so by the Chernoff bound, $|D'| \le 2.04s = 612\epsilon^{-2}$ except with probability $e^{-1.02s/3} < 10^{-40}$. Finally, a union bound over the three estimates, the sampler of Lemma B.15 with failure probability $0.001$, the Chebyshev step, and the size bound gives failure probability at most $0.008$.

To analyze the runtime, note that the oracle $\tau$ can be queried in $O(kd)$ time, so $M$ and $\widetilde{\mathrm{cost}}^{\Lambda}(A,x)$ can be computed in $\widetilde{O}(\sqrt{n}kd)$ time and $\widetilde{O}(\sqrt{n})$ queries by Lemmas H.11 and H.4. $\widetilde{n}_1, \ldots, \widetilde{n}_k$ can be estimated in $\widetilde{O}(\sqrt{n}k^{1.5}d)$ time and $\widetilde{O}(\sqrt{nk})$ queries. Finally, each sampling probability can be computed in $O(kd)$ time, so the time for the final sampling is $\widetilde{O}(\sqrt{ns} \cdot kd) = \widetilde{O}(\epsilon^{-1}n^{0.5}kd)$ with $\widetilde{O}(\epsilon^{-1}n^{0.5})$ queries. Computing the weights of the $O(\epsilon^{-2})$ sampled indices takes $O(\epsilon^{-2}kd)$ additional time, which is dominated since $\epsilon^{-1} \le n^{0.5}$ by our assumption that $n \gg \mathrm{poly}(k, 1/\epsilon)$. Thus, the overall runtime is

$$\widetilde{O}(n^{0.5}kd(\epsilon^{-1} + k^{0.5})). \qquad \square$$

Note that computing the weights classically would take $O(ndk)$ time, so ours is the first to achieve this goal in time sublinear in $n$. To compute a set of approximate $k$-centers, one could either use the bicriteria centers of Lemma H.2 as a proxy, or first compute an $\epsilon$-coreset using Theorem H.15 and then apply Lemma H.17 to the coreset.

# I Lower Bound

In this section, we prove a quantum query lower bound for the generic sensitivity sampling problem, showing that the $\widetilde{O}(\sqrt{ns})$ query complexity of Theorem C.3 is optimal up to $\mathrm{poly}(\epsilon^{-1}, \log n)$ and VC-dimension factors. Theorem I.1 uses the block construction of Theorem 8.1 of Apers and Gribling [2026] (arXiv v3), which shows that finding a constant-factor spectral approximation of $A^\top A$ requires $\Omega(\sqrt{nd})$ quantum row queries. For the dependence on $\epsilon$, Theorems I.3 and I.4 have a precedent in Xue et al. [2023]. Their Theorem 5.1 (restated as their Theorem D.5) proves that multidimensional counting with $m$ cells, the problem of Lemma H.5, requires $\Omega(\sqrt{nm}\,\epsilon^{-1/2})$ queries, by a reduction from the counting problems of Nayak and Wu [1999] on blocks, and the coreset part of their Theorem 5.2 (their Theorem D.7) derives from it the bound $\Omega(\sqrt{nk}\,\epsilon^{-1/2})$ for $\epsilon$-coresets for $(k,p)$-clustering. The leverage-score lower bound quoted in Section 2.3, Lemma 8.2 of Apers and Gribling [2026], is credited there to Theorem 2 of Apers and de Wolf [2022], and its hypotheses $n = O(d^2)$ and $\epsilon = \Omega(\sqrt{d/n})$ force $S = d = \Omega(\epsilon^{-2})$.

**Theorem I.1** (Query lower bound for generic sensitivity sampling)**.** *Let $n, S$ be positive integers with $S \le n$, and let $\epsilon \in (0, 1/2)$. There exists a family of instances, consisting of a point set $A = \{a_1, \ldots, a_n\} \subset \mathbb{R}^2$, a universe $X$ with $|X| = S$, and a cost function $\mathrm{cost} : \mathbb{R}^2 \times X \to \mathbb{R}_{\ge 0}$, such that the total sensitivity of every instance is at most $S$, the dimension of the system (Definition C.5) is $O(1)$, and any quantum algorithm that outputs an $\epsilon$-coreset of $A$ with probability at least $2/3$ requires $\Omega(\sqrt{nS})$ quantum queries to the points of $A$, provided the output coreset has size $o(\sqrt{nS})$.*

*Proof.* Without loss of generality let $S$ divide $n$, and partition $[n]$ into $S$ blocks $\mathcal{B}_1, \ldots, \mathcal{B}_S$, each of size $n/S$. The instance is specified by hidden bits $z \in \{0,1\}^n$: the $i$-th point is $a_i = (j(i), z_i) \in \mathbb{R}^2$,

where $j(i)$ denotes the block containing $i$, so that one query to a point reveals its bit. Let $X = \{x_1, \dots, x_S\}$ and define, for a point $a = (j, z)$,

$$\mathrm{cost}(a, x_l) = z \cdot \mathbf{1}[j = l],$$

so that $\mathrm{cost}(A, x_l)$ is the number of *marked* points (points with $z_i = 1$) in block $l$. The sensitivity of a marked point in block $l$ equals the inverse of the number of marked points in block $l$, and unmarked points have sensitivity 0. Hence, the total sensitivity is the number of nonempty blocks, which is at most $S$. Moreover, for a subset $M \subseteq [n]$ and a weight vector $v$, every range $\mathrm{range}_{M,v}(x_l, r)$ in the sense of Definition C.5 is $M$ with the marked points $i \in M \cap \mathcal{B}_l$ of weighted cost $v_i > r$ removed. All other points of $M$ have cost 0 under $x_l$, and hence they always belong to the range. Therefore, $M$ has at most $|M| + 1 \le |M|^2$ distinct ranges, and the system has $O(1)$ dimension.

Now let $B$ be a valid $\epsilon$-coreset of $A$ with $\epsilon < 1/2$. For every block $l$ with at least one marked point, $\mathrm{cost}(A, x_l) \ge 1$ and thus $\sum_{b \in B} w_b \cdot \mathrm{cost}(b, x_l) \ge (1 - \epsilon) > 0$, and since only marked points of block $l$ contribute to this sum, $B$ must contain a marked point of block $l$. Conversely, if block $l$ contains no marked point, then neither does $B \cap \mathcal{B}_l$. Therefore, reading the bits of the coreset points (at most $|B|$ additional queries) determines the entire vector $(\mathrm{OR}(z_{\mathcal{B}_1}), \dots, \mathrm{OR}(z_{\mathcal{B}_S}))$, where $\mathrm{OR}(z_{\mathcal{B}_l})$ denotes the disjunction of the bits in block $l$. Computing all $S$ disjunctions simultaneously, over independent instances of size $n/S$ each, requires $\Omega(S\sqrt{n/S}) = \Omega(\sqrt{nS})$ bounded-error quantum queries by the strong direct product theorem for OR [Klauck et al., 2007]. Hence, if the algorithm makes $Q$ queries and outputs a coreset of size $|B|$, then $Q + |B| = \Omega(\sqrt{nS})$, and the claim follows from $|B| = o(\sqrt{nS})$. □

**Remark I.2.** *As shown in Section 2.3, the coreset produced by Theorem C.3 for this instance family has size $s = \widetilde{O}(\epsilon^{-2}S)$, which is $o(\sqrt{nS})$ whenever $S = \widetilde{o}(\epsilon^4 n)$, and hence Theorem I.1 matches the $\widetilde{O}(\sqrt{ns})$ query complexity of our framework up to $\mathrm{poly}(\epsilon^{-1}, \log n)$ factors. We address the dependence on $\epsilon$ for this family in Theorems I.3 and I.4 below.*

**Theorem I.3** ($\epsilon$-dependence of the query lower bound, $S = 1$)**.** *Let $n$ be a positive integer and $\epsilon \in (0, 1/2)$ with $\lfloor 1/\epsilon \rfloor \le n/2$. There is a family of instances with a point set $A = \{a_1, \dots, a_n\} \subset \mathbb{R}^2$, a universe $X$ with $|X| = 1$ and a cost function as in Theorem I.1 (total sensitivity at most 1, dimension $O(1)$), such that any quantum algorithm that, on every instance of the family, outputs with probability at least $2/3$ an $\epsilon$-coreset of $A$ of size $o(\sqrt{n/\epsilon})$ together with its weight vector makes $\Omega(\sqrt{n/\epsilon})$ quantum queries to the points of $A$ on some instance. The bound is tight for this family: an $\epsilon$-coreset of size 1 can be output with $O(\sqrt{n/\epsilon})$ queries.*

*Proof.* We take $S = 1$ in the family of Theorem I.1, that is, $a_i = (1, z_i)$ for hidden bits $z \in \{0, 1\}^n$, $X = \{x_1\}$, and $\mathrm{cost}(a, x_1) = z$ for $a = (1, z)$. Thus, $\mathrm{cost}(A, x_1) = t := |z|$ is the number of marked points, and, as in Theorem I.1, the total sensitivity is at most 1 and the dimension is $O(1)$.

*Lower bound.* Let $(B, w)$ be the weighted $\epsilon$-coreset output by the algorithm, where the weights are part of the output since $B$ alone carries only membership information. Then $\sum_{b \in B} w_b \cdot \mathrm{cost}(b, x_1) = (1 \pm \epsilon) \cdot t$, and after reading the bits of the points of $B$, which takes at most $|B|$ further queries, the left-hand side is known. Therefore, an algorithm that outputs an $\epsilon$-coreset with $Q$ queries yields, with $Q + |B|$ queries, an estimate $\widetilde{t}$ with $|\widetilde{t} - t| \le \epsilon t$. In other words, it solves the $2\epsilon$-approximate relative counting problem of Nayak and Wu [1999] for $z$, and this problem is defined there with the strict inequality $|\widetilde{t} - t| < 2\epsilon t$. By Theorem 1.13 of Nayak and Wu [1999], an algorithm that solves the $\epsilon'$-approximate relative counting problem on all inputs with at least one marked point makes, on inputs with $t$ ones,

$$\Omega\Big(\sqrt{\frac{n}{\lceil \epsilon'(t+1) \rceil}} + \frac{\sqrt{t(n-t)}}{\lceil \epsilon'(t+1) \rceil}\Big)$$

queries. Note that for $t = 0$ the strict inequality cannot be met, and it is not required. The family consists of all $z \in \{0, 1\}^n$, so the algorithm must succeed on every input. We evaluate the bound with $\epsilon' = 2\epsilon$ on inputs with $t$ marked points, where $t + 1 = \lfloor 1/\epsilon \rfloor$, and for these inputs $\lceil 2\epsilon(t+1) \rceil \le 2$. Moreover, $t \ge 1$ as $\epsilon < 1/2$, and $t \le n/2$ as $\lfloor 1/\epsilon \rfloor \le n/2$, so the bound is $\Omega(\sqrt{t(n-t)}) = \Omega(\sqrt{tn}) = \Omega(\sqrt{n/\epsilon})$. Therefore, $Q + |B| = \Omega(\sqrt{n/\epsilon})$, and $Q = \Omega(\sqrt{n/\epsilon})$ whenever $|B| = o(\sqrt{n/\epsilon})$.

*Upper bound.* We compute an estimate $\widetilde{t}$ with $|\widetilde{t} - t| \le \epsilon t$ in two stages. First, the relative counter of Theorem 15 of Brassard et al. [2002] with accuracy $1/2$ returns $t_r = (1 \pm 1/2)t$ with probability at least $2/3$ in $O(\sqrt{n/t}) = O(\sqrt{n})$ expected queries, and $t_r = 0$ with certainty when $t = 0$. If $t_r \le 3/\epsilon$, then $t \le 2t_r \le 6/\epsilon$, and exact counting by Theorem 17 of Brassard et al. [2002] returns $t$ with probability at least $2/3$ in $O(\sqrt{(t+1)n}) = O(\sqrt{n/\epsilon})$ expected queries. Otherwise, $t \ge 2t_r/3 > 2/\epsilon$, and we run the relative counter with accuracy $\epsilon$, which returns $\widetilde{t} = (1 \pm \epsilon)t$ with probability at least $2/3$ in $O(\epsilon^{-1}\sqrt{n/t}) = O(\sqrt{n/\epsilon})$ expected queries. If $\widetilde{t} = 0$, we output the empty coreset. Otherwise, we find one marked point $b$ by the Grover search of Theorem 3 of Brassard et al. [2002] in $O(\sqrt{n/t}) = O(\sqrt{n})$ expected queries, and output $B = \{b\}$ with weight $w_b = \widetilde{t}$. Then we have $\sum_{b \in B} w_b \cdot \mathrm{cost}(b, x_1) = \widetilde{t} = (1 \pm \epsilon) \cdot \mathrm{cost}(A, x_1)$, so $B$ is an $\epsilon$-coreset of size 1. Each stage is repeated $O(1)$ times to make its error probability at most $1/20$, taking the median of the count estimates and checking a candidate point with one query. By Markov's inequality, each expected query bound becomes a worst-case bound by stopping the stage at 20 times the bound. This uses $O(\sqrt{n/\epsilon})$ queries in total, and a union bound over the stages gives success probability at least $2/3$. □

**Theorem I.4** ($\epsilon$-dependence of the query lower bound, general $S$)**.** *Let $S \ge 2$ be an integer (the case $S = 1$ is Theorem I.3, and nothing below uses $S \ge 2$), $\epsilon \in (0, 1/8]$ and $n$ a multiple of $S$ with $\lfloor 1/(4\epsilon) \rfloor \le n/(2S)$. On the block family of Theorem I.1 (total sensitivity at most $S$, dimension $O(1)$), any quantum algorithm that, on every instance, outputs with probability at least $2/3$ an $\epsilon$-coreset of size $o(\sqrt{nS/\epsilon})$ together with its weight vector makes $\Omega(\sqrt{nS/\epsilon})$ quantum queries to the points of $A$ on some instance. The bound is tight up to a $\log S$ factor: $O(\sqrt{nS/\epsilon}\log S)$ queries suffice on this family.*

*Proof.* Let $N := n/S$ be the block size and $t := \lfloor 1/(4\epsilon) \rfloor$, so we have $2 \le t \le N/2$, $\epsilon t \le 1/4$, and $t \ge 1/(8\epsilon)$. Note that $\epsilon \le 1/8$ gives $1/(4\epsilon) \ge 2$, hence $\lfloor 1/(4\epsilon) \rfloor \ge 1/(8\epsilon)$. Let $f := f_{t,t-1}$ be the partial Boolean function of Nayak and Wu [1999] on $\{0,1\}^N$. It is defined on inputs with exactly $t$ or exactly $t-1$ ones by $f(x) = 1$ if and only if $|x| = t$. We apply Corollary 1.2 of Nayak and Wu [1999] with $\Delta_\ell = 1$ and $m = t - 1$, which is the element of $\{t-1, t\}$ farthest from $N/2$ since $t \le N/2$. The corollary shows that every quantum algorithm computing $f$ with error at most a constant below $1/2$ makes $\Omega(\sqrt{N} + \sqrt{(t-1)(N-t+1)}) = \Omega(\sqrt{tN}) = \Omega(\sqrt{N/\epsilon})$ queries, as $t - 1 \ge t/2$, $N - t + 1 > N/2$, and $t \ge 1/(8\epsilon)$. We write $Q_{1/4}(f)$ for the $1/4$-error query complexity, which is of the same order.

*Reduction.* Consider an instance of the block family in which block $l$ has $t_l \in \{t-1, t\}$ marked points, and let $(B, w)$ be a weighted $\epsilon$-coreset output by the algorithm. As in the proof of Theorem I.3, we read the bits of the points of $B$ with at most $|B|$ queries. This yields, for every block $l$, a number $\widetilde{t}_l = \sum_{b \in B \cap \mathcal{B}_l} w_b\, z_b$ with $(1-\epsilon)t_l \le \widetilde{t}_l \le (1+\epsilon)t_l$. Note that $(1+\epsilon)(t-1) < (1-\epsilon)t$ if and only if $2\epsilon t < 1 + \epsilon$, and the latter holds as $\epsilon t \le 1/4$. Therefore, whenever the coreset is valid, the rule "output 1 if $\widetilde{t}_l \ge (1-\epsilon)t$" decides $f(z_{\mathcal{B}_l})$ correctly for every block simultaneously. Consequently, the algorithm, followed by $|B|$ further queries, computes $f^{(S)}(z_{\mathcal{B}_1}, \ldots, z_{\mathcal{B}_S}) = (f(z_{\mathcal{B}_1}), \ldots, f(z_{\mathcal{B}_S}))$ with success probability at least $2/3$ on every input of the product domain, using $Q + |B|$ queries.

*Direct product.* We write $Q_\eta(g)$ for the minimum number of queries of a quantum algorithm computing $g$ with error at most $\eta$ on every input of its domain. Theorem 1.1 of Lee and Roland [2013] states that for any partial function $f$, any integer $k \ge 1$, and any $\delta \in [2/3, 1]$, we have $Q_{1-\delta^k/2}(f^{(k)}) \ge \frac{k \ln(3\delta/2)}{8000}\, Q_{1/4}(f)$. We apply the theorem with $k = S$ and $\delta = 1$ to the algorithm above, which computes $f^{(S)}$ with error at most $1/3 \le 1/2$. Therefore, we have $Q + |B| \ge Q_{1/2}(f^{(S)}) \ge \frac{S \ln(3/2)}{8000}\, Q_{1/4}(f) = \Omega(S\sqrt{N/\epsilon}) = \Omega(\sqrt{nS/\epsilon})$, and $Q = \Omega(\sqrt{nS/\epsilon})$ whenever $|B| = o(\sqrt{nS/\epsilon})$.

*Upper bound.* We run the $S = 1$ algorithm of Theorem I.3 on each block, amplified to error $1/(3S)$ by $O(\log S)$ repetitions, and output the union of the $S$ single-point coresets. As in that proof, the amplification takes the median of the count estimates and one check query per candidate point. This uses $O(\sqrt{nS/\epsilon}\log S)$ queries in total, and a union bound gives success probability at least $2/3$. □

## J Simulating Random Oracle Access

In this section, we discuss how to simulate query access to a uniformly random string without impairing the runtime of our algorithm. We observe that this is necessary, as our classical oracle is inherently randomized. Converting it into a quantum circuit will produce a superposition of estimates, incurring additional approximation errors. We resolve this issue by "derandomizing" the algorithm, giving it query access to a uniformly random string. Specifically, suppose an algorithm has runtime $q$, then we need to attach a random string of length at most $q$. As observed in Apers and Gribling [2026], if we only care about query complexity, then we could explicitly sample these bitstrings and use QRAM to access them. In addition, if we want to achieve a sublinear runtime, then explicitly drawing these bitstrings would be too slow. On the other hand, Beals et al. [2001] shows that any $(k/2)$-query quantum algorithm cannot distinguish a uniformly random string from a $k$-wise independent one, hence it is enough to use a $k$-wise independent hash function to simulate query access to uniformly random strings. The following result is due to Apers and de Wolf [2022] and elaborated in Apers and Gribling [2026]:

**Lemma J.1** (Claim 1 in Apers and de Wolf [2022], restated in Lemma 3.12 of Apers and Gribling [2026])**.** *Consider any quantum algorithm with runtime $q$ that uses a uniformly random string, then we can simulate this algorithm with a quantum algorithm with runtime $\widetilde{O}(q)$ without random string, using an additional QRAM of $\widetilde{O}(q)$ bits.*

Hence, to implement our algorithm in QRAM, we apply Lemma J.1 to derandomize the procedure. This allows us to use more standard approaches to deal with deterministic routines: we convert them to quantum oracles using quantum arithmetic gates and Toffoli gates, which is common in the quantum algorithm literature. Each oracle in our applications reads its classical data, such as sketches, factors, and centers, from QRAM, adds up one term at a time, finds minima by binary search, and uncomputes all intermediate values, at a polylogarithmic overhead in time and queries. Hence, besides a register holding the queried row, column, or fiber, as in Apers and Gribling [2026], Xue et al. [2023], it uses $O(\log n)$ qubits, or polylogarithmically many if it calls the oracle of Lemma H.3, provided each kernel evaluation also uses $O(\log n)$ qubits. The exceptions are the oracle of Algorithm 9, which holds $\widetilde{O}(k/\epsilon)$ kernel values in a register, and the multidimensional counting and summation of Xue et al. [2023] in Lemmas H.5 and H.6, whose numbers of qubits we do not bound.

## K Query Complexity of Our Algorithms

In this section, we summarize the complexity of our algorithms in terms of *query complexity*, which is independent of the QRAM model. As noted in Apers and de Wolf [2022], Apers and Gribling [2026], we can remove the QRAM assumption at a price in *time complexity*. In particular, access to a QRAM of $m$ bits can be simulated in the circuit model at a multiplicative cost of $O(m)$ per access, see the discussion in Section 2 of Apers and Gribling [2026]. Our algorithms use a QRAM of $m = \widetilde{\Omega}(Q)$ bits, where $Q \geq n^{0.5}$ is the query count in Table 4. The $k$-wise independent string of Section J must fool $Q$ queries, so $k \geq 2Q$ [Beals et al., 2001], and hence the string alone occupies $\Omega(Q)$ bits. Moreover, each of the $Q$ queries is accompanied by at least one access to the string, and therefore this simulation yields a time bound of at least $\widetilde{\Omega}(Q \cdot m) = \widetilde{\Omega}(Q^2) \geq \widetilde{\Omega}(n)$. It is worth noting that this bound is a statement about the simulation, not about the problems. Whether sublinear time without QRAM is achievable for these problems is open. The query complexities below are independent of QRAM.

Table 4: Query complexity of our algorithms. Throughout, we assume $n, d \gg \text{poly}(k, 1/\epsilon)$. In all clustering and $\ell_p$ regression rows, $\widetilde{O}(\cdot)$ is $\widetilde{O}_p(\cdot)$. The $(k, 1 \le p < 2)$-subspace row assumes the flatness precondition of Theorem F.2. We specify the types of query to measure the complexity for various problems. For most of the problems, the input is $A \in \mathbb{R}^{n \times d}$ that consists of $n$ points in $d$-dimensional Euclidean space, hence the row query corresponds to querying the points. For low-rank approximation, the input is $A \in \mathbb{R}^{d \times n}$ and the queries are column queries, except that the factored output of Theorem D.8 also queries rows. For PSD and kernel low-rank approximation, the query corresponds to evaluating the kernel function over two points, where in the PSD low-rank approximation, this is equivalent to querying the entries of $A$. Finally, for tensor low-rank approximation, we allow queries to the rows, columns and tubes of $A$, which would require at most a constant factor more bits to store in the QRAM.

| | Reference | Query Type | Complexity |
|---|---|---|---|
| $k$-Median Clustering | Theorem H.15 | Rows | $n^{0.5}(k + \epsilon^{-1}k^{2/3})$ |
| $k$-Means Clustering | Theorem H.15 | Rows | $n^{0.5}(k + \epsilon^{-1}k^{3/4})$ |
| $(k, p)$-Clustering | Theorem H.15 | Rows | $n^{0.5}(k + \epsilon^{-1}k^{\frac{p+1}{p+2}})$ |
| $\ell_{p\le 2}$ Regression | Theorems C.18 and C.12 | Rows | $\epsilon^{-1}n^{0.5}d^{0.5}$ |
| $\ell_{p>2}$ Regression | Theorem C.18 | Rows | $n^{0.5}d^{p/4}(\epsilon^{-1} + d^{0.5})$ |
| $(k, 1 \le p < 2)$-Subspace Approx. | Theorem F.2 | Rows | $\epsilon^{-1}n^{1-p/4}k^{p/4}$ |
| $(k, p \ge 2)$-Subspace Approx. | Theorem F.2 | Rows | $\epsilon^{-2/p}n^{1-1/p}k^{0.5}$ |
| Low-Rank (column coreset) | Corollary D.3 | Columns | $\epsilon^{-1}n^{0.5}k^{0.5}$ |
| Low-Rank (factored output) | Theorem D.8 | Columns+Rows | $\epsilon^{-2}n^{0.5}k^{0.5} + \epsilon^{-1.5}d^{0.5}k^{0.5}$ |
| PSD Low-Rank | Theorem E.3 | Entries | $\epsilon^{-1.25}n^{0.75}k^{1.25}$ |
| Kernel Low-Rank | Theorem E.3 | Kernel Evals | $\epsilon^{-1.25}n^{0.75}k^{1.25}$ |
| Tensor Low-Rank: Rank-$k$ | Theorem G.7 | Rows, Columns and Tubes | $\epsilon^{-2}nk$ |
| Tensor Low-Rank: Bicriteria (rank $k^2$) | Theorem G.4 | Rows, Columns and Tubes | $\epsilon^{-1}nk^{0.5} + \epsilon^{-3}n^{0.5}k^2$ |

## 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: This paper is theoretical. The claimed bounds are summarized in Tables 1 and 2, Section 2 gives the proof overview, and full proofs are in Appendices C–I.

   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: We clearly discuss the limitations and three open problems in Section 3.

   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: We provide the full set of assumptions and proofs. An overview can be found in Section 2, and all details are in the appendix.

   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: [N/A]

   Justification: This paper is theoretical and has no experiments.

   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: [N/A]

   Justification: This paper is theoretical and has no data and code.

   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: [N/A]

   Justification: This paper is theoretical and has no experiments.

   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: [N/A]

   Justification: This paper is theoretical and has no experiments.

   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: [N/A]

   Justification: This paper is theoretical and has no experiments.

   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 strictly follow the Code of ethics of NeurIPS.

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: Our work is theoretical, and the quantum hardware it relies on still requires many further developments to realize. Moreover, it is also foundational research and is not tied to any applications or deployments. Hence, we foresee no negative societal impact, and also don't feel compelled to discuss it in the body of the paper.

    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: This paper is theoretical and has no experiments.

    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: [N/A]

    Justification: This paper is theoretical and has no experiments, hence does not use any existing assets.

    Guidelines:

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

13. **New assets**

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

    Answer: [N/A]

    Justification: This paper is theoretical and has no experiments and does not create any new assets.

    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: This paper is theoretical and has no experiments.

    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: This paper is theoretical and has no experiments.

    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: [Yes]

    Justification: LLMs were used as research assistants that helped establish several of the tighter results in this paper, including the lower bounds, the $\ell_p$ regression results, and the tighter bounds for $(k, p)$-clustering and $(k, p)$-subspace approximation. The authors verified all proofs and take full responsibility for the content, and the acknowledgments state how LLMs were used.

    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.