Near-Optimal Quantum Coreset Construction Algorithms for Clustering

Yecheng Xue [*1] **Xiaoyu Chen** [*2] **Tongyang Li** [1] **Shaofeng H.-C. Jiang** [1]

Abstract

k-Clustering in $\mathbb{R}^d$ (e.g., k-median and k-means) is a fundamental machine learning problem. While near-linear time approximation algorithms were known in the classical setting for a dataset with cardinality n, it remains open to find sublinear-time quantum algorithms. We give quantum algorithms that find coresets for k-clustering in $\mathbb{R}^d$ with $\tilde{O}(\sqrt{nk}d^{3/2})$ query complexity. Our coreset reduces the input size from n to $\mathrm{poly}(k\varepsilon^{-1}d)$, so that existing α-approximation algorithms for clustering can run on top of it and yield $(1+\varepsilon)\alpha$-approximation. This eventually yields a quadratic speedup for various k-clustering approximation algorithms. We complement our algorithm with a nearly matching lower bound, that any quantum algorithm must make $\Omega(\sqrt{nk})$ queries in order to achieve even $O(1)$-approximation for k-clustering.

1. Introduction

Clustering is a fundamental machine learning task that has been extensively studied in areas including computer science and operations research. A typical clustering problem is k-median clustering in $\mathbb{R}^d$. In k-median clustering, we are given a set of data points $D \subseteq \mathbb{R}^d$ and an integer parameter k, and the goal is to find a set $C \subset \mathbb{R}^d$ of k points, called the center set, such that the following objective is minimized:

$$\mathrm{cost}(D, C) := \sum_{x \in D} \mathrm{dist}(x, C), \tag{1}$$

here $\mathrm{dist}(x, y) := \|x - y\|_2$, $\mathrm{dist}(x, C) := \min_{c \in C} \mathrm{dist}(x, c)$.

In the classical setting, k-median clustering is shown to be NP-hard (Megiddo & Supowit, 1984) even in the planar case (i.e., Euclidean $\mathbb{R}^2$), and polynomial time approximation algorithms were the focus of study. Furthermore, even allowing $O(1)$-approximation, a fundamental barrier is still that the algorithm must make $\Omega(nk)$ accesses to the distances between the n input points (Mettu & Plaxton, 2004).

In this paper, we study quantum algorithm and complexity for k-median clustering (and more generally, (k, z)-clustering as in Definition 2.1). This is motivated by quantum algorithms for various related data analysis problems, such as classification (Kapoor et al., 2016; Li et al., 2019; 2021), nearest neighbor search (Wiebe et al., 2015), support vector machine (Rebentrost et al., 2014), etc. Many of these quantum algorithms are originated from the Grover algorithm (Grover, 1996), which can find an item in a data set of cardinality n in time $O(\sqrt{n})$, a quadratic quantum speedup compared to the classical counterpart. Hence, a natural question is whether quantum algorithms can break the aforementioned classical $\Omega(nk)$ lower bound for k-median clustering, with a practical goal of achieving complexity $O(\sqrt{nk})$, while still achieving $O(1)$ or even $(1+\varepsilon)$-approximation.

Coresets To this end, we consider constructing coresets (Har-Peled & Mazumdar, 2004; Feldman & Langberg, 2011) by quantum algorithms. The coreset is a powerful technique for dealing with clustering problems. Roughly speaking, an ε-coreset is a tiny proxy of the potentially huge data set, such that the clustering cost on any center set is preserved within ε relative error. For k-median, an ε-coreset of size $\mathrm{poly}(k\varepsilon^{-1})$ has been known to exist (see e.g., Sohler & Woodruff, 2018; Huang & Vishnoi, 2020; Braverman et al., 2021; Cohen-Addad et al., 2021; Cohen-Addad et al., 2022; Schwiegelshohn et al., 2022), which is independent of both the dimension d and size n of the data set. Once such a coreset is constructed, one can approximate k-means efficiently by plugging in existing approximation algorithms (so that the input size is reduced to the size of the coreset which is only $\mathrm{poly}(k\varepsilon^{-1})$). In addition, coresets can also be applied to clustering algorithms in sublinear settings such as streaming (Har-Peled & Mazumdar, 2004), distributed computing (Balcan et al., 2013), and dynamic algorithms (Henzinger & Kale, 2020).

[*]Equal contribution [1]Center on Frontiers of Computing Studies, Peking University, Beijing, China [2]School of Electronics Engineering and Computer Science, Peking University, Beijing, China. Correspondence to: Shaofeng H.-C. Jiang <shaofeng.jiang@pku.edu.cn>, Tongyang Li <tongyangli@pku.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

Contributions We propose the first quantum algorithm for constructing coresets for k-median that runs in $\tilde{O}\left(\sqrt{nk}\right)$ time,[1] which breaks the fundamental linear barrier of classical algorithms.

Theorem 1.1 (Informal version of Theorem 3.1)**.** *There exists an quantum algorithm that given $\varepsilon > 0$ and an n-point data set $D \subset \mathbb{R}^d$, returns an ε-coreset of size $\tilde{O}\left(kd\,\mathrm{polylog}(n)/\varepsilon^2\right)$ for k-median over D with success probability at least $2/3$, with query complexity $\tilde{O}\left(\sqrt{nk}d^{3/2}/\varepsilon\right)$ and additional $\mathrm{poly}\left(kd\log n/\varepsilon\right)$ processing time.*

Our coreset construction also yields coresets of a similar size for the related k-median clustering problem (and more generally, (k, z)-clustering, see Definition 2.1), using the same order of query and processing time (see Theorem 3.1). The size bound of our coreset stated in Theorem 1.1 may not be optimal, but since it already has size $\mathrm{poly}(k\varepsilon^{-1}d)$, one can trivially apply the state-of-the-art classical coreset construction algorithm on top of our coreset to obtain improved bounds. For instance, we can obtain coresets for k-means of size $O\left(k\varepsilon^{-4}\right)$ using Cohen-Addad et al. (2022), and an alternative size bound of $O\left(k^{1.5}\varepsilon^{-2}\right)$ using Schwiegelshohn et al. (2022). These require the same order of query and processing time as in Theorem 1.1.

In addition, by a similar argument, our coreset readily implies efficient approximation algorithms for clustering problems. In particular, one constructs an ε-coreset S as in Theorem 1.1 and applies an existing α-approximate algorithm with input S, then it yields an $O((1+\varepsilon)\alpha)$-approximation to the original problem. The query complexity of the entire process remains the same as in Theorem 1.1, and it only incurs additional $T(\mathrm{poly}(k\varepsilon^{-1}\log n))$ processing time, provided that the approximation algorithm runs in $T(n)$ time for an n-point dataset. This particularly implies a quantum PTAS (for fixed d) for k-median and k-means using $\tilde{O}\left(\sqrt{nk}d^{3/2}/\varepsilon\right)$ queries, and $\mathrm{poly}(k)\cdot f(d,\varepsilon)$ processing time for some function f of d and ε, by applying Cohen-Addad et al. (2021).

Our quantum algorithms (and lower bounds) for clustering are summarized in Table 1.

Techniques The general idea of our algorithm is to quantize and combine two existing algorithms, the approximate algorithm by (Thorup, 2005) and a recent seminal coreset construction algorithm by Cohen-Addad et al. (2021).

In a high level, we start with computing a bicriteria solution which uses slightly more than k points to achieve $O(1)$-approximation to OPT. This step is based on (Thorup, 2005). Then given this solution, we partition the dataset D into groups, perform a sampling procedure in each group, and re-weight the sampled points to form the coreset, following the idea in (Cohen-Addad et al., 2021).

For the bicriteria approximation, we provide a quantum implementation for the algorithm by Thorup (2005) in Section 3.1 to obtain a solution that has $O(k\,\mathrm{poly}\log n)$ points with cost being $O(1)$ multiple of OPT. A key step in this algorithm is to query for the nearest neighbor of each data point $x \in D$ in a given point set with size $O(k)$. This step is straightforward in the classical setting with cost $O(nk)$ by calculating the exact nearest neighbor and store the results. However, to achieve the $\tilde{O}\left(\sqrt{nk}\right)$ complexity in the quantum setting, we need improvements on nearest neighbor search. We make use of a well-known approximate nearest neighbor search technique called *locality sensitive hashing* (LSH), and the version that we use gives $2(1+\varepsilon)$-approximate nearest neighbor using $N^{\mathrm{poly}(1/\varepsilon)}$ preprocessing time and $\tilde{O}(d)\cdot\mathrm{poly}(1/\varepsilon)$ query time for N points in $\mathbb{R}^d$ (Indyk & Motwani, 1998). In Section 3.2, we also use our quantum implementation of LSH (Lemma 3.4) to construct a unitary that encodes the clusters induced by the approximate solution, i.e., maps each $x \in D$ to a corresponding center.

After we obtain a bicriteria approximation A in the previous step, we adapt the algorithm in (Cohen-Addad et al., 2021) to build the coreset, where the dataset D is partitioned into $\tilde{O}(z^2/\varepsilon^2)$ groups with respect to A. In this partition procedure, a key step is to calculate $\mathrm{cost}(C_i, A) = \sum_{x\in C_i}\mathrm{dist}^z(x, A)$ for each cluster C_i induced by A. While this seems simple in the classical setting where one directly computes the cost of each data point and summing up over each cluster in $O(nk)$ time, this task is nontrivial in quantum since we aim for sublinear complexity. To design a sublinear quantum algorithm that approximately computes the cost of all clusters simultaneously, we propose a new subroutine called multidimensional quantum approximate summation (Theorem 4.2). Specifically, given an oracle $O : |i\rangle|0\rangle|0\rangle \to |i\rangle|\tau(i)\rangle|f(i)\rangle$, where $\tau\colon [n]\to[m]$ is a partition and $f\colon [n]\to\mathbb{R}_{\geq 0}$ is a bounded function, Theorem 4.2 shows that using in total $\tilde{O}\left(\sqrt{nm/\varepsilon}\right)$ queries to O_τ one can obtain ε-estimation of $\sum_{\tau(i)=j} f(i)$ for each $j \in [m]$. This algorithm may be of independent interest.

To complement our algorithm results, we also prove quantum lower bounds for approximate k-means clustering. We consider three settings in which an ε-coreset, an ε-optimal set of centers, or an ε-estimate of the optimal clustering cost are outputted, and we prove $\Omega\left(\sqrt{nk}\varepsilon^{-1/2}\right), \Omega\left(\sqrt{nk}\varepsilon^{-1/6}\right)$ and $\Omega\left(\sqrt{nk}+\sqrt{n}\varepsilon^{-1/2}\right)$ lower bounds, respectively. These

[1] In this paper, we use $\tilde{O}$ to omit poly-logarithmic terms in O.

Table 1. Classical and quantum complexity bounds for clustering in $\mathbb{R}^d$ (omitting dependence in d). Note that an $O(1)$-coreset for k-median (resp. k-means) implies an $O(1)$-approximate solution for k-median (resp. k-means).

Reference	Type	Problem	Time Complexity
Braverman et al. (2017)	Classical upper bound	ε-coreset for k-median	$\tilde{O}(n+\mathrm{poly}(k))$
Theorem 3.1	Quantum algorithm	ε-coreset for k-median	$\tilde{O}(\sqrt{nk}/\varepsilon)$
Theorem 3.1	Quantum algorithm	ε-coreset for k-means	$\tilde{O}(\sqrt{nk}/\varepsilon)$
Theorem 5.2	Quantum lower bound	$O(1)$-approximate k-median	$\Omega(\sqrt{nk})$
Theorem 5.2	Quantum lower bound	$O(1)$-approximate k-means	$\Omega(\sqrt{nk})$

quantum lower bounds confirm that our quantum algorithms for clustering problems are *near-optimal in n and k*, up to a logarithmic factor. In general, we start with proving the quantum lower bounds when $k = 1$ by reducing from the approximate quantum counting problem (Nayak & Wu, 1999). We then obtain the general bounds with $\sqrt{k}$ factors by applying composition theorems (Theorem D.1) in a refined manner. We speculate that the gap between different settings might be intrinsic, which is also discussed in a recent paper (Charikar & Waingarten, 2022).

Related work In general, quantum algorithms for machine learning are of general interest (Biamonte et al., 2017; Schuld & Petruccione, 2018; Dunjko & Briegel, 2018). We compare our results to existing literature in quantum machine learning as follows.

Aïmeur et al. (2007) conducted an early study on quantum algorithms for clustering, including divisive clustering, k-median clustering, and neighborhood graph construction. Their k-median algorithm has complexity $O\left(n^{3/2}/\sqrt{k}\right)$, which is at least n and slower than our quantum algorithm.

Lloyd et al. (2013) gave a quantum algorithm for cluster assignment and cluster finding with complexity $\mathrm{poly}(\log nd)$. However, their quantum algorithm requires the input data to be sparse with efficient access to nonzero elements, i.e., each of $x_1, \ldots, x_n$ has $\mathrm{poly}(\log d)$ nonzero elements and we can access these coordinates in $\mathrm{poly}(\log d)$ time. In addition, their algorithm outputs quantum states instead of classical vectors. More caveats are listed in Aaronson (2015).

The most relevant result is Kerenidis et al. (2019), which gave an quantum algorithm named q-means for k-means clustering. Q-means has complexity $\tilde{O}(k^2 d\eta^{2.5}\varepsilon^{-3} + k^{2.5}\eta^2\varepsilon^{-3})$ per iteration for well-clusterable datasets, where η is a scaling factor for input data such that $1 \le \|x_i\|_2^2 \le \eta$ for all $i \in [n]$. For general datasets the complexity is larger and depends on condition number parameters. Q-means can be extended to spectral clustering (Kerenidis & Landman, 2021). As a comparison, our quantum algorithm does not require the well-clusterable assumption (nor condition numbers related to this) and can be regarded as a direct speedup of common classical algorithms for k-means. In addition, our quantum algorithm only has $\mathrm{poly}(\log \eta)$ dependence, and the dependence on k and $1/\varepsilon$ is also better.

There are also heuristic quantum machine learning approaches for clustering (Otterbach et al., 2017; Poggiali et al., 2022) and other problems in data analysis (Schuld et al., 2017; Farhi & Neven, 2018; Kerenidis & Luongo, 2018; Havlíček et al., 2019). These results do not have theoretical guarantees at the moment, and we look forward to their further developments on heuristic performances and provable guarantees. In addition, Chia et al. (2022) proposed a quantum-inspired classical algorithm for 1-mean clustering with sampling access to input data.

Open questions Our work leaves several natural open questions for future investigation:

- Can we give fast quantum coreset construction algorithms for other related clustering problems with complexity $\tilde{O}\left(\sqrt{nk}\right)$? Potential problems include fair clustering (Chierichetti et al., 2017), capacitated clustering (Cohen-Addad & Li, 2019; Braverman et al., 2022), k-center clustering (Agarwal & Procopiuc, 2002), etc.

- Can we give fast quantum coreset construction algorithms for clustering problems in more general metric spaces? Note that Cohen-Addad et al. (2021) gives the result for various metric space, such as doubling metrics, graphs, and general discrete metric spaces, while still achieving $\tilde{O}(nk)$ time in the classical setting. However, to achieve this similar coreset size bound using time $O(\sqrt{nk})$ in the quantum setting seems nontrivial; for instance, one cannot make use of the LSH technique that we use to speed up the approximate nearest neighbor search.

2. Preliminaries

2.1. Notations

We give the notations and definitions used in the following. In this paper, we focus on the *(k, z)-clustering* problem in an Euclidean space:

Definition 2.1 ((k, z)-Clustering)**.** Given a data set $D \subset \mathbb{R}^d$, $z \geq 1$ and integer $k \geq 1$, the (k, z)-clustering problem is to find a set $C \subset \mathbb{R}^d$ with size k, minimizing the cost function

$$\mathrm{cost}(D, C) := \sum_{x \in D} (\mathrm{dist}(x, C))^z .$$

Here $\mathrm{dist}(x, y) := ||x - y||_2$, $\mathrm{dist}(x, C) := \min_{c \in C} \mathrm{dist}(x, c)$.

For $z = 1$ it is also known as the *k-median* problem, and for $z = 2$ it is also known as the *k-means* problem. Any set $C \subset \mathbb{R}^d$ with size k can be seen as a *solution*. Given a solution C, a point $c \in C$ is a *center*. We can map each data point $x \in D$ to a certain $c \in C$, and the set $\{x \in D : x \text{ is mapped to } c\}$ is a *cluster*. Let OPT be the cost of the optimal solution. We assume the distance is rescaled so that the minimum intra-point distance is 1, and we assume the diameter of the point set is $\mathrm{poly}(n)$. Hence, $\mathrm{OPT} = \mathrm{poly}(n)$.

An important related concept is the *coreset*:

Definition 2.2 (Coreset)**.** Given a data set $D \subset \mathbb{R}^d$, $z \geq 1$ and integer $k \geq 1$, a weighted set S with weight function $w \colon S \to \mathbb{R}_+$ is called an ε-coreset if

$$\forall C \subset \mathbb{R}^d, |C| \leq k, \;\; \mathrm{cost}(S, C) \in (1 \pm \varepsilon) \cdot \mathrm{cost}(D, C)$$

where $\mathrm{cost}(S, C) = \sum_{s \in S} w(s) \cdot \mathrm{dist}^z(s, C)$.

As a special case, we call S unweighted if $w(s) = 1 \; \forall s \in S$.

In this paper, we refer to an ε-estimation for a number p by $\tilde{p}$ if $|\tilde{p} - p| \leq \varepsilon p$. We use $[n]$ for $\{1, \ldots, n\}$.

2.2. Basics of Quantum Computing

The basic unit of a classical computer is a bit, and in quantum computing it is a *qubit*. Mathematically, a system of m qubits forms an M-dimensional Hilbert space for $M = 2^m$. Any *quantum state* $|\phi\rangle$ in this space can be written as

$$|\phi\rangle = \sum_{i=0}^{M-1} \alpha_i |i\rangle , \;\; \text{where} \;\; \sum_{i=0}^{M-1} |\alpha_i|^2 = 1. \qquad (2)$$

Here $\{|0\rangle, \ldots, |M-1\rangle\}$ forms an orthonormal basis in the Hilbert space called as the *computational basis*, and $\alpha_i \in \mathbb{C}$ is called as the *amplitude* of $|i\rangle$. Intuitively, the quantum state $|i\rangle$ can be regarded as a classical state i, and the quantum state $|\phi\rangle$ in (2) is a *superposition* of classical states. The operations in quantum computing are *unitaries* matrices following the principles of linear algebra. Specifically, a unitary acting on an M-dimensional Hilbert space can be formulated as follows:

$$U_f \colon \; |i\rangle |0\rangle \to |i\rangle |f(i)\rangle , \; \forall i \in \{0, \ldots, M-1\}.$$

Note that due to linearity, U_f works not only for the basis vectors $\{|i\rangle\}_{i=0}^{M-1}$, but also for any quantum state in this Hilbert space. For example, by applying U_f to $|\phi\rangle$ we obtain the following quantum state

$$|\phi\rangle = \sum_{i=0}^{M-1} \alpha_i |i\rangle \xrightarrow{U_f} \sum_{i=0}^{M-1} \alpha_i |f(i)\rangle .$$

This allows us to perform calculations "in parallel" and achieve the quantum speedup.

Quantum access to the input data is also unitary and can be encoded as a *quantum oracle*. We state the definitions of the *probability oracle* and the *binary oracle* as follows:

Definition 2.3 (Probability Oracle)**.** Let $p \colon [M] \to \mathbb{R}_{\geq 0}$ be a probability distribution. We say O_p is a probability oracle for p if

$$O_p \colon \; |0\rangle \to \sum_{j \in [M]} \sqrt{p(j)} \, |j\rangle \, |\phi_j\rangle ,$$

where $|\phi_j\rangle$ are arbitrary ℓ_2-normalized vectors.

Definition 2.4 (Binary Oracle)**.** Let $D = \{x_1, \ldots, x_n\}$ be a subset of $\mathbb{R}^d$. We say O_D is a binary oracle for D if

$$O_D \colon \; |i\rangle |0\rangle \to |i\rangle |x_i\rangle , \; \forall i \in [n].$$

The definition of the binary oracle also fits for any vector $w = (w_1, \ldots, w_N) \in \mathbb{R}^N$. Binary oracle is a common input model in quantum algorithms and we also call the binary oracle of D (or w) as the quantum query to D (or w). Besides, for two point sets $S \subset D \subset \mathbb{R}^d$, we say O_S is the membership query to S if

$$O_S \colon \; |x\rangle |0\rangle \to |x\rangle |I(x \in S)\rangle , \; \forall x \in D$$

where $I(x \in S)$ is the indicator for whether $x \in S$.

In a quantum algorithm, we can also write information to a *quantum-readable classical-writable classical memory* (QRAM) and make it encoded as an oracle (Giovannetti et al., 2008). We refer *query complexity* as the number of queries to the input oracle and the QRAM. *Time complexity* is referred as the total processing time, including all the use of queries, quantum gates, and classical operations.

2.3. Quantum Speedup

Here, we introduce basic problems which can be sped-up by quantum computing. Those tools are rudimentary to our quantum algorithms as well as others in machine learning.

Quantum Sampling In our quantum algorithms, we the following quantum sampling algorithm:

Lemma 2.5 (Rephrased from Theorem 1 of Hamoudi 2022)**.** *There is a quantum algorithm such that: given two integers $1 \leq m \leq n$, a real $\delta > 0$, and a non-zero vector $w \in \mathbb{R}^n_{\geq 0}$, with a probability at least $1 - \delta$, the algorithm outputs a sample set S of size m such that each element $i \in [n]$ is sampled with probability proportional to w_i using $O(\sqrt{nm} \log(1/\delta))$ quantum queries to w in expectation.*

Quantum Counting and Search In quantum computing, counting the number of points satisfying a specific property can be solved with quadratic speedup:

Lemma 2.6 (Theorem 15 of Brassard et al. 2002)**.** *There is a quantum algorithm such that given a real $\delta > 0$ and two sets $S \subset D \subset \mathbb{R}^d$, $|S| = m$, $|D| = n$, it outputs an ε-estimation $\tilde{m}$ for m with probability at least $1 - \delta$ using $O(\varepsilon^{-1}\sqrt{n/m}\log(1/\delta))$ queries to D and membership queries to S.*

Furthermore, quantum speedup can also be achieved with outputting all such points, known as repeated Grover search:

Lemma 2.7 (Claim 2 of Apers & de Wolf 2020)**.** *There is a quantum algorithm such that given a real $\delta > 0$ and two sets $S \subset D \subset \mathbb{R}^d$, $|S| = m$, $|D| = n$, it finds S with probability at least $1 - \delta$ using $\tilde{O}(\sqrt{nm}\log(1/\delta))$ queries to D and membership queries to S.*

Quantum Sum Estimation Beyond counting and search, quadratic quantum speedup can also be achieved for estimating the sum of a set of numbers:

Lemma 2.8 (Rephrased from Lemma 6 of Li et al. 2019)**.** *Consider $D = \{x_1, \ldots, x_n\} \subset \mathbb{R}^n_{\geq 0}$ and denote $x = \sum_{i=1}^N x_i$ as the sum of all the elements in D. There is a quantum algorithm such that given $\delta > 0$, it outputs $\tilde{x}$ as an ε-estimation for x with probability at least $1 - \delta$, using $O(\sqrt{n}\log(1/\delta)/\varepsilon)$ queries to D.*

3. Coreset Construction

This section presents a quantum algorithm for coreset construction in $\tilde{O}(\sqrt{nk})$ time. This algorithm combines and quantizes two existing classical algorithms, the bicriteria approximation algorithm of Thorup (2005) and the coreset construction algorithm (based on an approximate solution) of Cohen-Addad et al. (2021). This paper focuses on the (k, z)-clustering problem (Definition 2.1) over a size-n data set $D = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$, and assumes the access to oracle $O_D\colon |i\rangle|0\rangle \to |i\rangle|x_i\rangle\ \forall i \in [n]$. The main result for coreset construction is as follows.

Theorem 3.1. *There exists a quantum algorithm such that given data set $D \subset \mathbb{R}^d$, positive real $\varepsilon < 1/2^{O(z)}$, $z \geq 1$, and integer $k \geq 1$, it returns an ε-coreset for (k, z)-clustering over D of size $\tilde{O}\left(2^{O(z)}kd\,\mathrm{polylog}(n)\max\left(\varepsilon^{-2}, \varepsilon^{-z}\right)\right)$ with success probability at least $2/3$ using:*

- $\tilde{O}\left(2^{O(z)}\sqrt{nkd}\max\left(\varepsilon^{-1}, \varepsilon^{-z/2}\right)\right)$ *queries to O_D,*
- $\tilde{O}\left(2^{O(z)}\sqrt{nk}d^{3/2}\max\left(\varepsilon^{-1}, \varepsilon^{-z/2}\right)\right)$ *queries to QRAM,*
- $\mathrm{poly}\left(kd\log n/\varepsilon^z\right)$ *additional processing time.*

Remark 3.2. When $d \geq \Omega(\log n/\varepsilon^2)$, one can apply the Johnson-Lindenstrauss transform (Johnson & Lindenstrauss, 1984) as a preprocessing step to obtain the following alternative bounds.

- $\tilde{O}\left(2^{O(z)}k\,\mathrm{polylog}(n)\max\left(\varepsilon^{-4}, \varepsilon^{-z-2}\right)\right)$ coreset size,
- $\tilde{O}\left(2^{O(z)}\sqrt{nk}\max\left(\varepsilon^{-2}, \varepsilon^{-z/2-1}\right)\right)$ queries to O_D,
- $\tilde{O}\left(2^{O(z)}\sqrt{nkd}\max\left(\varepsilon^{-4}, \varepsilon^{-z/2-3}\right)\right)$ queries to QRAM,
- $\mathrm{poly}\left(k\log n/\varepsilon^z\right) + O\left(d\log n/\varepsilon^2\right)$ additional processing time.

These bounds have tight asymptotic dependence in d.

The quantum algorithm contains two parts. First, Section 3.1 presents an algorithm to compute a bicriteria approximate solution A, which is an approximate solution with size slightly larger than k. Then, based on this A, Section 3.2 presents an algorithm for coreset construction. Combining the two algorithms directly yields Theorem 3.1.

For clustering problem it is a basic subroutine to find the nearest neighbor of each data point $x \in D$ in a given set $A \subset \mathbb{R}^d$ since the optimization objective is the cost function $\mathrm{cost}(D, A) = \sum_{x\in D}\mathrm{dist}^z(x, A)$ for any $A \subset \mathbb{R}^d$. It always holds that $|A| = k\,\mathrm{polylog}(n)$. This can be easily implemented in the classical setting, since the nearest neighbor of all the $x \in D$ can be found with $\tilde{O}(nk)$ time and all the information can be stored with $O(n)$ space. However, this approach cannot be easily adapted to yield the $\tilde{O}(nk)$ complexity in the quantum setting. In particular, the step of exactly computing the nearest center can require $\Omega(k)$ time. Hence, this paper uses a mapping that maps each point $x \in D$ to an approximately nearest neighbor in A instead. This paper takes the advantage of an existing classical result, which is based on a widely adopted technique, Locality Sensitive Hashing (LSH).

Lemma 3.3 (Approximate Nearest Neighbor Search, Rephrased from Theorem 2.10 and Theorem 3.17 of Indyk & Motwani 1998)**.** *There exists an algorithm such that given two parameters $\delta' > 0$, $\varepsilon \in (0, 1/2)$, for any set $A \subset \mathbb{R}^d$, $|A| = m$, it constructs a data structure using $m^{O(\log(1/\varepsilon)/\varepsilon^2)}\log(1/\delta')$ space and preprocessing time, such that for any query $x \in \mathbb{R}^d$, with probability at least $1 - \delta'$ it answers $a \in A$ which satisfies $\mathrm{dist}(x, a) \leq 2(1+\varepsilon)\,\mathrm{dist}(x, A)$ using $\varepsilon^{-2}d\,\mathrm{polylog}(m/\delta')$ query time.*

For a quantum version, there exists an algorithm that performs the preprocessing classical and stores the data structure in QRAM (Giovannetti et al., 2008), and then answers queries in a quantum manner based on the stored information. This yields a quantum algorithm with the same

query time since any classical operation can be simulated by constant quantum operations. Let $\varepsilon = c/2 - 1$. Setting $\delta' = \delta/n$ and using the union bound yields the following lemma.

Lemma 3.4 (Quantum Approximate Nearest Neighbor Search)**.** *There exists an algorithm such that given two parameters $\delta > 0$, $c_\tau \in [5/2, 3)$, for any $A \subset \mathbb{R}^d$, $|A| = m$, it constructs an oracle*

$$O_\tau \colon |i\rangle |0\rangle \to |i\rangle |\tau(i)\rangle, \forall i \in [n]$$

using $\mathrm{poly}(m \log(n/\delta))$ classical preprocessing time and QRAM space. With success probability at least $1 - \delta$, $\tau \colon [n] \to [m]$ is a mapping such that

$$\mathrm{dist}(x_i, a_{\tau(i)}) \le c_\tau \, \mathrm{dist}(x_i, A) \quad \forall i \in [n].$$

Each query to O_τ uses $d \, \mathrm{polylog}(mn/\delta)$ queries to QRAM.

Remark 3.5. By the same method, one can construct the oracle to a mapping that maps any $i \in [n]$ to the corresponding center $a_{\tau(i)} \in A$ instead of the index $\tau(i)$, with the same complexity.

3.1. Bicriteria Approximation

For a (k, z)-clustering problem, its bicriteria approximate solution is defined as follows:

Definition 3.6. Assume that OPT is the optimal cost for the (k, z)-clustering problem. An (α, β)-bicriteria approximate solution is a point set $A \subset \mathbb{R}^d$ such that $|A| \le \alpha k$, $\mathrm{cost}(D, A) \le \beta \, \mathrm{OPT}$.

This section presents a quantum algorithm that finds a bicriteria solution with $\tilde{O}(d\sqrt{nk})$ query complexity, as stated in Lemma 3.7 below. This algorithm is a quantization of a classical algorithm by Thorup (2005).

Lemma 3.7. *Algorithm 1 outputs an $(O(\log^2 n), 2^{O(z)})$-bicriteria approximate solution A with probability at least $5/6$, using $\tilde{O}(\sqrt{nk})$ calls to O_D and its inverse, $\tilde{O}(d\sqrt{nk})$ queries to a QRAM, and $\mathrm{poly}(k \log n)$ additional processing time.*

$\tau_{t+1} \colon D \to A_{t+1}$ in Algorithm 1 Line 7 is a mapping such that for some constant c_τ

$$\mathrm{dist}(x, \tau_{t+1}(x)) \le c_\tau \, \mathrm{dist}(x, A_{t+1}).$$

It holds that $|A_{t+1}| = O(k \, \mathrm{polylog} \, n)$ for any t. Using Lemma 3.4, for any $c_\tau \in [5/2, 3)$, an oracle

$$O_{\tau_{t+1}} \colon |x\rangle |0\rangle \to |x\rangle |\tau(x)\rangle$$

can be constructed using $\mathrm{poly}(k \log(n))$ classical preprocessing time and QRAM space, and each query to $O_{\tau_{t+1}}$ uses $d \, \mathrm{polylog}(kn)$ queries to QRAM and constant query to O_D.

Algorithm 1 Bicriteria Approximation

1: **input:** k, z, n, oracle O_D
2: **output:** $(O(\log^2 n), 2^{O(z)})$-bicriteria approximation A
3: initialize $t \leftarrow 0$, $D_0 \leftarrow D$, $A_0 \leftarrow \emptyset$, $\tilde{r}_0 \leftarrow n$
4: **repeat**
5: draw a uniform sample S_t of size $13k\lceil \log n \rceil$ over D_t using Lemma 2.5. $A_{t+1} \leftarrow S_t \cup A_t$
6: draw a sample s_t uniformly at random over D_t using Lemma 2.5
7: construct a map $\tau_{t+1} \colon D \to A_{t+1}$
$d_{t+1} \leftarrow \mathrm{dist}(s_t, \tau_{t+1}(s_t))$
$D_{t+1} \leftarrow D_t \setminus \{x \in D : \mathrm{dist}(x, \tau_{t+1}(x)) \le d_{t+1}\}$
8: make an $1/2$-estimation $\tilde{r}_{t+1}$ for $r_{t+1} = |D_{t+1}|$ using Lemma 2.6
9: $t \leftarrow t + 1$
10: **until** $\tilde{r}_t \le 39k\lceil \log n \rceil$ or $t \ge 3\lceil \log n \rceil$
11: find all points in D_t using Lemma 2.7, $A \leftarrow D_t \cup A_t$
12: repeat all the steps above for three times and union all the A

Proof of Lemma 3.7. Algorithm 1 is a quantum implementation of Algorithm D of Thorup (2005), which constructs a set of size $O(k \log^2 n/\varepsilon)$ that contains a factor $2 + \varepsilon$ approximation to k-median problem for $\varepsilon \in (0, 1/2)$ with probability at least $1/2$. Algorithm 1 has small difference from the classical algorithm but it does not influence the correctness; a detailed proof is given in Appendix A.

In the classical setting, to identify the set D_t one can list all the points in it or in the dataset D make those points marked. In the quantum setting, to identify D_t is to construct the unitary

$$U_{D_t} \colon |x\rangle |0\rangle \to |x\rangle |I(x \in D_t)\rangle, \quad \forall x \in D,$$

where $I(x \in D_t)$ is the indicator for whether $x \in D_t$. This unitary can be constructed iteratively:

$$\begin{aligned} U_{D_{t+1}} \colon & |x\rangle |0\rangle |0\rangle |0\rangle \\ & \xrightarrow{U_{D_t}, O_{\tau_{t+1}}} |x\rangle |I(x \in D_t)\rangle |\tau_{t+1}(x)\rangle |0\rangle \\ & \mapsto |x\rangle |I(x \in D_t)\rangle |\tau_{t+1}(x)\rangle |I(x \in D_{t+1})\rangle \\ & \xrightarrow{O^{-1}_{\tau_{t+1}}, O^{-1}_{D_t}} |x\rangle |0\rangle |0\rangle |I(x \in D_{t+1})\rangle. \end{aligned}$$

For Line 5-6, Algorithm 1 applies Lemma 2.5 with unitary U_{D_t} and $m = 13k\lceil \log n \rceil + 1$, which uses $\tilde{O}(\sqrt{nk})$ calls for O_{D_t}, O_D, and their inverses. This algorithm uses $\tilde{O}(\sqrt{n})$ for Line 8 and $\tilde{O}(nk)$ for Line 11 queries to O_{D_t}, O_D, and their inverses. All the steps above are repeated for no more than $O(\log n)$ times, and it can be concluded that in total the algorithm uses $\tilde{O}(d\sqrt{nk})$ queries for a QRAM, $\tilde{O}(\sqrt{nk})$ queries for O_D and its inverse, and additional $\mathrm{poly}(k \log n)$ processing time.

We further note that the failure probability of our quantum algorithm gives at most a poly-logarithmic factor. In Algorithm 1, each subroutine in use suffers only a $\log(1/\delta)$ factor to reach a success probability at least $1-\delta$ and each of them is applied no more than $\mathrm{poly}(nk)$ times, so setting the failure probability as $\delta = O(1/\mathrm{poly}(nk))$ for each subroutine and the union bound ensures that all the applications to quantum subroutines success with high probability. This cause only a $\mathrm{polylog}(nk)$ factor and it is absorbed by the $\tilde{O}$ notation. □

3.2. Coreset Construction

We present a quantum algorithm for constructing a coreset based on a bicriteria approximate solution. The construction is a quantum implementation of Cohen-Addad et al. (2021). Algorithm 2 shows a sketch of the construction.

Algorithm 2 requires an access to an (α, β)-bicriteria approximation A, which means an oracle $O_A\colon |i\rangle|0\rangle \to |i\rangle|a_i\rangle$ $\forall i \in [m]$ for a set $A = \{a_1, \ldots, a_m\} \subset \mathbb{R}^d$, $m \le \alpha k$, $\mathrm{cost}(D, A) \le \beta\,\mathrm{OPT}$. Based on A, using Lemma 3.4, one can construct an oracle

$$O_\tau\colon |s\rangle|0\rangle \to |s\rangle|i\rangle, \quad \forall s \in [n]$$

where $\mathrm{dist}(x_s, a_i) \le c_\tau \,\mathrm{dist}(x_s, A)$ $\forall s \in [n]$ for some constant c_τ. This oracle encodes a mapping $\tau\colon [n] \to [m]$ which maps each $x_s \in D$ to an approximately nearest neighbor $a_i \in A$ as its center. The map τ together with A can be seen as a special solution for clustering. Let $\mathrm{cost}_\tau(D', A) := \sum_{x\in D'} \mathrm{dist}^z(x, a_{\tau(x)})$ be the cost of this solution and let $C_i := \{x \in D \mid \tau(x) = i\}$ as the i-th cluster induced by τ and A for any $i \in [m]$.

Lemma 3.8. *Let*

$$t = \tilde{O}\left(2^{O(z)} \cdot m \cdot (d + \log(n)) \cdot \max\left(\varepsilon^{-2}, \varepsilon^{-z}\right)\right)$$

in Algorithm 2. For a positive real $\varepsilon < 1/(4c_\tau^z)$, Algorithm 2 outputs an $O(c_\tau^z \beta\varepsilon)$-coreset of size $\tilde{O}\left(2^{O(z)} md\log(n)\max\left(\varepsilon^{-2}, \varepsilon^{-z}\right)\right)$ with probability at least $5/6$, using $\tilde{O}\left(2^{O(z)} c_\tau \sqrt{nmd}\max\left(\varepsilon^{-1}, \varepsilon^{-z/2}\right)\right)$ queries to O_τ, O_D, O_A, their inverses, and QRAM. Besides it uses $\mathrm{poly}(md\log n/\varepsilon^z)$ additional classical processing time.

The details of Algorithm 2 are described as follows. This algorithm consists of two phases. During the first phase the algorithm partitions the dataset D into groups. This consists of two steps. The first step of the first phase is to partition each cluster into rings, with each ring containing the points with the same distance from the center up to factor 2. For each C_i, let

$$R_{i,j} := \{x \in C_i \mid 2^j \Delta_{C_i} \le \mathrm{cost}_\tau(x, A) \le 2^{j+1}\Delta_{C_i}\}.$$

Algorithm 2 Coreset Construction

input: t, ε, oracle O_D, O_A, O_τ
output: $O(c_\tau^z\beta\varepsilon)$-coreset Ω with weight function w
// phase 1: partitioning the dataset into groups
compute $\Delta_{C_i} = \mathrm{cost}_\tau(C_i, A)/|C_i|$ for each $i \in [m]$ by using O_τ and Theorem 4.2, store in QRAM
construct the ring unitary U_R
compute $\mathrm{cost}_\tau(R_{i,j}, A)$ and $\mathrm{cost}_\tau(R_j, A)$ for each pair i, j by using U_R and Theorem 4.2, store in QRAM
construct the group unitary U_G
// phase 2: sensitivity sampling and reweighting
compute $\mathrm{cost}_\tau(G_{j,b}, A)$ for each pair of j and b using U_G and Theorem 4.2, store in QRAM
for each $i \in [m]$, compute $|R_{i,I}| + |C_i \cap (\cup_{j\neq I} G_{j,\min})|$ by Theorem 4.4, and assign the value to $w(a_i)$
for each well-structured group G, draw a size-t i.i.d. sample Ω such that each $x \in C_i \cap G$ is selected in each round with the same probability

$$\Pr[x] = \frac{\mathrm{cost}_\tau(C_i, A)}{|C_i|\,\mathrm{cost}_\tau(G, A)}$$

using Lemma 2.5 and reweight $w(x) = 1/(t\Pr[x])$ for each sampled point x
for each outer group G, draw a size-t i.i.d. sample Ω such that each each $x \in G$ is selected in each round with probability

$$\Pr[x] = \frac{\mathrm{cost}_\tau(x, A)}{\mathrm{cost}_\tau(G, A)}$$

using Lemma 2.5 and reweight $w(x) = 1/(t\Pr[x])$ for each sampled point x
let Ω be the union of all the above samples and A

Let $R_{i,I} := \cup_{j \le -2z\log(z/\varepsilon)} R_{i,j}$ be the inner ring and $R_{i,O} := \cup_{j > 2z\log(z/\varepsilon)}$ be the outer ring. Besides, let $R_I := \cup_{i=1}^m R_{i,I}$, $R_O := \cup_{i=1}^m R_{i,O}$, and

$$R_j := \cup_{i=1}^m R_{i,j} \quad \forall j,\ -2z\log(z/\varepsilon) < j \le 2z\log(z/\varepsilon).$$

The ring unitary U_R is defined as

$$U_R\colon |s\rangle|0\rangle|0\rangle \to |s\rangle|i\rangle|j\rangle \quad \forall s \in [n]$$

where $x_s \in R_{i,j}$ for each $s \in [n]$. For $j \le -2z\log(z/\varepsilon)$, U_R uses a same special notation for such j and in this paper it is denoted as $j = I$. The special notation can be any preselected value out of $[-2z\log(z/\varepsilon), 2z\log(z/\varepsilon)]$. And it is the same for $j = O$. U_R is a unitary which answers the corresponding ring $R_{i,j}$ for each query $x_s \in D$.

The second step is to gather the rings into groups such that the rings with equal cost up to factor 2 are gathered together and prepared to be handled together in the second phase.

For each $j \in \{\lceil -2z\log(z/\varepsilon)\rceil, \ldots, \lfloor 2z\log(z/\varepsilon)\rfloor + 1\} \cup \{I, O\}$, let

$$G_{j,b} := \cup_{i\in I_{j,b}} R_{i,j},$$

where $I_{j,b}$ is the largest set such that for any $i \in I_{j,b}$,

$$\mathrm{cost}_\tau(R_{i,j}, A) \in (\frac{\varepsilon}{4z})^z \cdot \frac{\mathrm{cost}_\tau(R_j, A)}{m} \cdot [2^b, 2^{b+1}].$$

And let $G_{j,\min} := \cup_{b\le 0} G_{j,b}$ be the union of the cheapest groups, and $G_{j,\max} := \cup_{b\ge z\log(4z/\varepsilon)} G_{j,b}$ be the union of the most expensive ones. The same notation as $j = I$ and $j = O$ is used for $b = \min$ and $b = \max$. The group unitary U_G is defined as

$$U_G\colon\ |s\rangle\,|0\rangle\,|0\rangle \to |s\rangle\,|j\rangle\,|b\rangle \quad \forall s \in [n],$$

where $x_s \in G_{j,b}$ for any $s \in [n]$. U_G answers the corresponding group $G_{j,b}$ for each query $x_s \in D$.

In the second phase the dataset seen as the union of three different kinds of points and these three parts are handle separately. The first kind contains the union of inner rings R_I and the cheapest groups $G_{j,\min}\ \forall j$; The second kind is all the well-structured groups $G_{j,b}$ with $-2z\log(z/\varepsilon) < j \le 2z\log(z/\varepsilon)$ and $b = 1, \ldots, \max$; And the third kind is all the outer groups $G_{O,b}$ with $b = 1, \ldots, \max$.

In quantum computing, it is costly to compute the exact sum such as $\mathrm{cost}_\tau(C_i, A)$ and $\mathrm{cost}_\tau(R_{i,j}, A)$. Hence, Algorithm 2 uses ε-estimations instead of corresponding exact values, but for convenience they are simply written as they are exact. It turns out that $O(\varepsilon)$-estimations are enough for constructing an $O(\varepsilon)$-coreset.

Proof of Lemma 3.8. The proof is shown in Appendix B. □

Remark 3.9. We note that in Line 3 (and similarly, Line 5 and 7), computing Δ_{C_i} for all the m clusters in $\tilde{O}(\sqrt{nm})$ time is feasible in quantum computing. We can compute $\{\mathrm{cost}_\tau(C_i, A)\}_{i=1}^m$ and $\{|C_i|\}_{i=1}^m$ separately, and then calculate the division in a classical method, so we only state for the calculation of all the $\mathrm{cost}_\tau(C_i, A)$. We can construct the following unitary U by one query to O_τ and its inverse.

$$U\colon\ |s\rangle\,|0\rangle\,|0\rangle \to |s\rangle\,|\tau(s)\rangle\,|\mathrm{dist}^z(x_s, A)\rangle \quad \forall s \in [n]$$

Note that $\mathrm{cost}_\tau(C_i, A) = \sum_{\tau(s)=i} \mathrm{dist}^z(x_s, A)$. According to Theorem 4.2, calculating these values requests only $\tilde{O}(\sqrt{nm})$ time. More details about Theorem 4.2 is shown in Section 4.

Let $m = O(k\log n)$, $\beta = 2^{O(z)}$, $\varepsilon = \varepsilon'/2^{O(z)}$, and $c_\tau = 5/2$ in Lemma 3.8. Note that O_A is obtained by storing A in QRAM, and one query to O_τ uses $d\,\mathrm{polylog}(mn)$ queries to QRAM by Lemma 3.4. Combining Lemma 3.7 and Lemma 3.8 directly yields Theorem 3.1.

4. Multidimensional Approximate Summation

A crucial subroutine of Algorithm 2 is to compute the summation of the $\mathrm{cost}_\tau(x, A)$ over all the points x in each part for a given partition (Line 3, 5, and 7). This gives rise to such a problem:

Definition 4.1 (Multidimensional Approximate Summation)**.** Given two integers $1 \le m \le n$, a real parameter $\varepsilon > 0$, a partition $\tau\colon [n] \to [m]$, and a function $f\colon [n] \to \mathbb{R}_{\ge 0}$. The multidimensional approximate summation problem is to find ε-estimation for each $s_j := \sum_{\tau(i)=j} f(i)$, $j \in [m]$.

This paper proposes *multidimensional quantum approximate summation* to solve this problem. We believe this technique can have wide applications in designing quantum algorithms for machine learning and other relevant problems.

Theorem 4.2 (Multidimensional Quantum Approximate Summation)**.** *Assume that there exists access to an oracle $O_\tau\colon\ |i\rangle\,|0\rangle\,|0\rangle \to |i\rangle\,|\tau(i)\rangle\,|f(i)\rangle\ \forall i \in [n]$ and assume that f has an upper bound M. For $\varepsilon \in (0, 1/3)$, $\delta > 0$, there exists a quantum algorithm that solves the multidimensional approximate summation problem with probability at least $1-\delta$, using $\tilde{O}\left(\sqrt{nm/\varepsilon}\log(1/\delta)\log M\right)$ queries to O_τ, $\tilde{O}\left((\sqrt{nm/\varepsilon} + m/\varepsilon)\log(n/\delta)\log M\right)$ gate complexity, and additional $O(m\log M)$ classical processing time.*

$f(i)$ is a binary number of length $\lceil \log M\rceil$. By first computing the summation of each digit and then summing up the results together, the multidimensional approximate summation problem can be reduced to the following problem:

Definition 4.3 (Multidimensional Counting)**.** Given two integers $1 \le m \le n$, a real parameter $\varepsilon > 0$, and a partition $\tau\colon [n] \to [m]$. For each $j \in [m]$, denote $D_j := \{i \in [n] : \tau(i) = j\}$ as the j-th part and $n_j := |D_j|$ for the size. The multidimensional counting problem is to find ε-estimation $\tilde{n}_j$ for each n_j, $j \in [m]$.

For this problem, this section proposes *multidimensional quantum counting* to solve it.

Theorem 4.4 (Multidimensional Quantum Counting)**.** *Assume that there exists access to an oracle $O_\tau\colon\ |i\rangle\,|0\rangle \to |i\rangle\,|\tau(i)\rangle\ \forall i \in [n]$. For $\varepsilon \in (0, 1/3)$, $\delta > 0$, Algorithm 3 solves the multidimensional counting problem with probability at least $1-\delta$, using $\tilde{O}\left(\sqrt{nm/\varepsilon}\log(1/\delta)\right)$ queries to O_τ and additional $\tilde{O}\left((\sqrt{nm/\varepsilon} + m/\varepsilon)\log(n/\delta)\right)$ gate complexity. The query complexity is optimal up to a logarithm factor.*

Denote $p_j := n_j/n$. Note that by one call for O_τ one can

construct the unitary

$$O_p\colon\ |0\rangle \to \sum_{j=1}^{m} \sqrt{p_j}|j\rangle \left(\frac{1}{n_j}\sum_{i\in D_j}|i\rangle\right). \tag{3}$$

Consider $p = (p_1, \ldots, p_m)$ as an m-dimensional probability distribution, the above unitary can be seen as a probability oracle to p. The following method is used to estimate the distribution:

Lemma 4.5 (Multidimensional Amplitude Estimation, Rephrased from Theorem 5 and Lemma 7 of van Apeldoorn 2021)**.** *There is a quantum algorithm which has the following properties: given precision $\varepsilon \in (0, 1/3)$, error probability $\delta > 0$, a quantum probability oracle O_p on q qubits for an m-dimensional probability distribution p, a number set $S \subset [m]$, and a constant $p_{mt} \geq \sum_{i\in S} p_i$ as the maximal total probability on S, it outputs $\tilde{p} \in \mathbb{R}^m$ such that*

$$|\tilde{p}_i - p_i| \leq \varepsilon \quad \forall i \in S$$

with probability $\geq 1-\delta$ using $O\left(\varepsilon \log(m/\delta)\sqrt{p_{mt}\varepsilon}\right)$ calls to O_p and membership queries to S. The gate complexity is $\tilde{O}\left((q(m + p_{mt}/\varepsilon) + \sqrt{p_{mt}}/\varepsilon)\log(1/\delta)\right)$ using a QRAM.

A naive method is to set $p_{mt} = 1$ and apply Lemma 4.5 directly. However, it requests an ε very small to ensure the size estimation of the small parts sufficiently precise, which leads to gratuitous overprecise estimation for the large parts and results in the waste of time. Algorithm 3 performs a trick to obtain the estimations hierarchically: in each iteration it sets a certain precision (Line 6), and saves only the estimation values large enough to ensure the accuracy and leaves the small parts to be estimated more precisely next time (Line 8-10).

The proof of Theorem 4.4 and Theorem 4.2 is deferred to Appendix C.

Remark 4.6. Note that our Algorithm 3 for multidimensional quantum computing is optimal up to a logarithmic factor due to Theorem 5.1.

Algorithm 3 Multidimensional Quantum Counting

1: **input:** $n, m, O_\tau, \varepsilon \in (0, \frac{1}{3}), \delta > 0$
2: **output:** $\{\tilde{n}_1, \cdots, \tilde{n}_m\}$, s.t. $|\hat{n}_j - n_j| \leq \varepsilon n_j\ \forall j \in [m]$
3: initialize $P \leftarrow [m]$, $Q \leftarrow [n]$, $\tilde{n} \leftarrow n$, $p_{mt} \leftarrow 1$
4: construct the oracle O_p in (3)
5: **repeat**
6: for each $j \in P$, estimate $\{p_j\}$ as $\{\tilde{p}_j\}$ by applying Lemma 4.5 with $q = \lceil \log n \rceil$, precision $\frac{2\tilde{n}\varepsilon}{3nm}$, maximal total probability p_{mt}, error probability $O(\frac{\delta}{\log n})$
7: **for** $j \in P$ **do**
8: **if** $\tilde{p}_j \geq \frac{\tilde{n}}{9nm}$ **then**
9: $\tilde{n}_j \leftarrow n\tilde{p}_j$, $P \leftarrow P - \{j\}$
10: **end if**
11: **end for**
12: $Q \leftarrow \{i \in [n] : \tau(i) \in P\}$
13: make an $1/2$-estimation $\tilde{n}$ for $|Q|$ using Lemma 2.6
14: $p_{mt} \leftarrow \frac{2\tilde{n}}{n}$
15: **until** $\tilde{n} < m/\varepsilon$
16: find all the items in Q using Lemma 2.7, count the remaining parts classically.

5. Lower Bound

To complement our quantum algorithms, we also prove the following quantum lower bounds. First, we have:

Theorem 5.1 (Quantum Lower Bound for Multidimensional Counting)**.** *Every quantum algorithm that solves the multidimensional counting problem (Definition 4.3) w.p. at least $\frac{2}{3}$ uses at least $\Omega\left(\sqrt{nk}\varepsilon^{-1/2}\right)$ queries to O_τ.*

The proof of Theorem 5.1 is deferred to Appendix D.2. For the clustering problems, we establish the following lower bounds under different settings (proofs deferred to Appendix D.3).

Theorem 5.2 (Quantum Lower Bounds for k-means and k-median)**.** *Assume that ε is sufficiently small. Consider the Euclidean k-means/median problem on data set $D = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$. Assume a quantum oracle $O_x|i, b\rangle := |i, b \oplus x_i\rangle$. Then, every quantum algorithm outputs the followings with probability $2/3$ must have quantum query complexity lower bounds for the following problems:*

- *An ε-coreset: $\Omega\left(\sqrt{nk}\varepsilon^{-1/2}\right)$ for k-means and k-median (Theorem D.7);*
- *An ε-estimation to the value of the objective function: $\Omega\left(\sqrt{nk} + \sqrt{n}\varepsilon^{-1/2}\right)$ for k-means and k-median (Theorem D.8);*
- *A center set C such that $\mathrm{cost}(C) \leq (1+\varepsilon)\,\mathrm{cost}(C^*)$ where C^* is the optimal solution: $\Omega\left(\sqrt{nk}\varepsilon^{-1/6}\right)$ for k-means; $\Omega\left(\sqrt{nk}\varepsilon^{-1/3}\right)$ for k-median (Theorem D.9).*

Acknowledgements

This paper is partially supported by a national key R&D program of China No. 2021YFA1000900 and a startup fund from Peking University.

References

Aaronson, S. Read the fine print. *Nature Physics*, 11(4): 291, 2015.

Agarwal, P. K. and Procopiuc, C. M. Exact and approxi-

mation algorithms for clustering. *Algorithmica*, 33(2): 201–226, 2002.

Aïmeur, E., Brassard, G., and Gambs, S. Quantum clustering algorithms. In *International Conference on Machine Learning*, pp. 1–8, 2007.

Apers, S. and de Wolf, R. Quantum speedup for graph sparsification, cut approximation and Laplacian solving. In *Proceedings of the 61st Annual Symposium on Foundations of Computer Science*, pp. 637–648. IEEE, 2020. arXiv:1911.07306

Balcan, M., Ehrlich, S., and Liang, Y. Distributed k-means and k-median clustering on general communication topologies. In *Advances in Neural Information Processing Systems*, pp. 1995–2003, 2013. arXiv:1306.060

Beals, R., Buhrman, H., Cleve, R., Mosca, M., and de Wolf, R. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. arXiv:quant-ph/9802049

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., and Lloyd, S. Quantum machine learning. *Nature*, 549(7671):195, 2017. arXiv:1611.09347

Brassard, G., Høyer, P., Mosca, M., and Tapp, A. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002. arXiv:quant-ph/0005055

Braverman, V., Frahling, G., Lang, H., Sohler, C., and Yang, L. F. Clustering high dimensional dynamic data streams. In *International Conference on Machine Learning*, pp. 576–585. PMLR, 2017. arXiv:1706.03887

Braverman, V., Jiang, S. H., Krauthgamer, R., and Wu, X. Coresets for clustering in excluded-minor graphs and beyond. In *Proceedings of the 32nd ACM-SIAM Symposium on Discrete Algorithms*, pp. 2679–2696. SIAM, 2021. arXiv:2004.07718

Braverman, V., Cohen-Addad, V., Jiang, H.-C. S., Krauthgamer, R., Schwiegelshohn, C., Toftrup, M. B., and Wu, X. The power of uniform sampling for coresets. In *Proceedings of the 63rd Annual Symposium on Foundations of Computer Science*, pp. 462–473. IEEE, 2022. arXiv:2209.01901

Charikar, M. and Waingarten, E. Polylogarithmic sketches for clustering. In *Proceedings of the 49th International Colloquium on Automata, Languages, and Programming*, volume 229 of *LIPIcs*, pp. 38:1–38:20. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2022. arXiv:2204.12358

Chia, N.-H., Gilyén, A. P., Li, T., Lin, H.-H., Tang, E., and Wang, C. Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning. *Journal of the ACM*, 69(5):1–72, 2022. arXiv:1910.06151

Chierichetti, F., Kumar, R., Lattanzi, S., and Vassilvitskii, S. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*, pp. 5029–5037, 2017. arXiv:1802.05733

Cohen-Addad, V. and Li, J. On the fixed-parameter tractability of capacitated clustering. In *46th International Colloquium on Automata, Languages, and Programming*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 41:1–41:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. arXiv:2208.14129

Cohen-Addad, V., Feldmann, A. E., and Saulpic, D. Near-linear time approximation schemes for clustering in doubling metrics. *Journal of the ACM*, 68(6):44:1–44:34, 2021. arXiv:1812.08664

Cohen-Addad, V., Saulpic, D., and Schwiegelshohn, C. A new coreset framework for clustering. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 169–182, 2021. arXiv:2104.06133

Cohen-Addad, V., Larsen, K. G., Saulpic, D., and Schwiegelshohn, C. Towards optimal lower bounds for k-median and k-means coresets. In Leonardi, S. and Gupta, A. (eds.), *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1038–1051. ACM, 2022. arXiv:2202.12793

Dunjko, V. and Briegel, H. J. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, 81(7):074001, 2018. arXiv:1709.02779

Farhi, E. and Neven, H. Classification with quantum neural networks on near term processors. *arXiv preprint*, 2018. arXiv:1802.06002

Feldman, D. and Langberg, M. A unified framework for approximating and clustering data. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, pp. 569–578, 2011. arXiv:1106.1379

Giovannetti, V., Lloyd, S., and Maccone, L. Quantum random access memory. *Physical Review Letters*, 100(16): 160501, 2008. arXiv:0708.1879

Grover, L. K. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, pp. 212–219. ACM, 1996. arXiv:quant-ph/9605043

Gupta, A., Krauthgamer, R., and Lee, J. R. Bounded geometries, fractals, and low-distortion embeddings. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pp. 534–543. IEEE, 2003.

Hamoudi, Y. Preparing many copies of a quantum state in the black-box model. *Physical Review A*, 105(6):062440, 2022. arXiv:2207.11014

Har-Peled, S. and Mazumdar, S. On coresets for k-means and k-median clustering. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pp. 291–300. ACM, 2004. arXiv:1810.12826

Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., and Gambetta, J. M. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 2019. arXiv:1804.11326

Henzinger, M. and Kale, S. Fully-dynamic coresets. In *Proceedings of the 28th Annual European Symposium on Algorithms*, volume 173 of *LIPIcs*, pp. 57:1–57:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. arXiv:2004.14891

Høyer, P., Lee, T., and Spalek, R. Negative weights make adversaries stronger. In Johnson, D. S. and Feige, U. (eds.), *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pp. 526–535. ACM, 2007. arXiv:quant-ph/0611054

Huang, L. and Vishnoi, N. K. Coresets for clustering in Euclidean spaces: importance sampling is nearly optimal. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1416–1429. ACM, 2020. arXiv:2004.06263

Indyk, P. and Motwani, R. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 604–613, 1998.

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

Kapoor, A., Wiebe, N., and Svore, K. Quantum perceptron models. In *Advances in Neural Information Processing Systems*, pp. 3999–4007, 2016. arXiv:1602.04799

Kerenidis, I. and Landman, J. Quantum spectral clustering. *Physical Review A*, 103(4):042415, 2021. arXiv:2007.00280

Kerenidis, I. and Luongo, A. Quantum classification of the MNIST dataset via slow feature analysis. *arXiv preprint*, 2018. arXiv:1805.08837

Kerenidis, I., Landman, J., Luongo, A., and Prakash, A. q-means: A quantum algorithm for unsupervised machine learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. arXiv:1812.03584

Kimmel, S. Quantum adversary (upper) bound. *Chicago Journal of Theoretical Computer Science*, 2013: 4, 2013. URL http://cjtcs.cs.uchicago.edu/articles/2013/4/contents.html. arXiv:1101.0797

Lee, T., Mittal, R., Reichardt, B. W., Spalek, R., and Szegedy, M. Quantum query complexity of state conversion. In Ostrovsky, R. (ed.), *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science*, pp. 344–353. IEEE Computer Society, 2011. arXiv:1011.3020

Li, T., Chakrabarti, S., and Wu, X. Sublinear quantum algorithms for training linear and kernel-based classifiers. In *International Conference on Machine Learning*, pp. 3815–3824. PMLR, 2019. arXiv:1904.02276

Li, T., Wang, C., Chakrabarti, S., and Wu, X. Sublinear classical and quantum algorithms for general matrix games. *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 35(10):8465–8473, 2021. arXiv:2012.06519

Lloyd, S., Mohseni, M., and Rebentrost, P. Quantum algorithms for supervised and unsupervised machine learning. *arXiv preprint*, 2013. arXiv:1307.0411

Megiddo, N. and Supowit, K. J. On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1):182–196, 1984.

Mettu, R. R. and Plaxton, C. G. Optimal time bounds for approximate clustering. *Mach. Learn.*, 56(1-3):35–60, 2004. arXiv:1301.0587

Nayak, A. and Wu, F. The quantum query complexity of approximating the median and related statistics. In Vitter, J. S., Larmore, L. L., and Leighton, F. T. (eds.), *Proceedings of the 31st Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pp. 384–393. ACM, 1999. arXiv:quant-ph/9804066

Otterbach, J. S., Manenti, R., Alidoust, N., Bestwick, A., Block, M., Bloom, B., Caldwell, S., Didier, N., Schuyler Fried, E., Hong, S., Karalekas, P., Osborn, C. B., Papageorge, A., Peterson, E. C., Prawiroatmodjo, G., Rubin, N., Ryan, C. A., Scarabelli, D., Scheer, M., Sete, E. A., Sivarajah, P., Smith, R. S., Staley, A., Tezak, N., Zeng, W. J., Hudson, A., Johnson, B. R., Reagor, M., da Silva, M. P., and Rigetti, C. Unsupervised machine learning

on a hybrid quantum computer. *arXiv preprint*, 2017. arXiv:1712.05771

Poggiali, A., Berti, A., Bernasconi, A., Del Corso, G., and Guidotti, R. Quantum clustering with k-means: a hybrid approach. *arXiv preprint*, 2022. arXiv:2212.06691

Rebentrost, P., Mohseni, M., and Lloyd, S. Quantum support vector machine for big data classification. *Physical Review Letters*, 113(13):130503, 2014. arXiv:1307.0471

Reichardt, B. W. Span programs are equivalent to quantum query algorithms. *SIAM Journal on Computing*, 43(3): 1206–1219, 2014.

Schuld, M. and Petruccione, F. *Supervised learning with quantum computers*, volume 17. Springer, 2018.

Schuld, M., Fingerhuth, M., and Petruccione, F. Implementing a distance-based classifier with a quantum interference circuit. *Europhysics Letters*, 119(6):60002, 2017. arXiv:1703.10793

Schwiegelshohn, C., Saulpic, D., Larsen, K. G., Cohen-addad, V. P., and Sheikh-Omar, O. A. Improved coresets for Euclidean k-means. In *Advances in Neural Information Processing Systems*, 2022. arXiv:2211.08184

Sohler, C. and Woodruff, D. P. Strong coresets for k-median and subspace approximation: Goodbye dimension. In *Proceedings of the 59th Annual Symposium on Foundations of Computer Science*, pp. 802–813. IEEE Computer Society, 2018. arXiv:1809.02961

Thorup, M. Quick k-median, k-center, and facility location for sparse graphs. *SIAM Journal on Computing*, 34(2): 405–432, 2005.

van Apeldoorn, J. Quantum probability oracles & multidimensional amplitude estimation. In *16th Conference on the Theory of Quantum Computation, Communication and Cryptography*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

Wiebe, N., Kapoor, A., and Svore, K. M. Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *Quantum Information & Computation*, 15(3-4):316–356, 2015. arXiv:1401.2142

A. Further Proof Details for Bicriteria Approximation

This section gives a rigorous proof of the correctness of the bicriteria approximate algorithm (Lemma 3.7), that the output of Algorithm 1 (the set A) is an $(O(\log^2 n), 2^{O(z)})$-bicriteria approximate solution with probability at least $5/6$. This proof follows Thorup (2005).

First, the loop (Line 4-10) stops for $t < 3\lceil \log n \rceil$ with high probability. In each iteration, assume all the points $x \in D_t$ are sorted $\mathrm{dist}(x, \tau_t(x))$ from small to large. The sample s_t is drawn from D_t uniformly at random (Line 6) and all the points x preceding s_t are deleted from D_t (Line 7), so with probability at least $1/2$ it holds that $|D_{t+1}| \le |D_t|/2$. If there's still $\tilde{r}_t > 39k\lceil \log n \rceil$ after $3\lceil \log n \rceil$ iterations, it holds that $|D_t| \ge 2\tilde{r}_t/3 > 26k\lceil \log n \rceil \ge 1$. Hence the event $|D_{t+1}| \le |D_t|/2$ happens in no more than $\lceil \log n \rceil$ iterations, which is only $2/3$ of the expectation. The probability that such an event happens is at most $\exp\left(-(1.5\log n)(1/3)^2/2\right) = \exp\left(-(\log n)/12\right) < n^{-0.12}$.

In the following, it is supposed that the loop (Line 4-10) stops for $t < 3\lceil \log n \rceil$. In this situation, it holds that

$$|A| \le 13k\lceil \log n \rceil \cdot 3\lceil \log n \rceil + 2 \cdot 39\lceil \log n \rceil = O(k \log^2 n).$$

For every $x \in D_t$, let c_x be the corresponding center in the optimal solution. x is called "happy point" and this is denoted as $\mathrm{Happy}(x)$ if there exists $c \in A_{t+1}$ such that $\mathrm{dist}(c, c_x) \le \mathrm{dist}(x, c_x)$. Otherwise x is called "angry point" and this event is denoted as $\mathrm{Angry}(x)$. For a certain $x \in D$, suppose that there are q points in D_t corresponding to c_x in the optimal solution and as close to c_x as x. Since in A_{t+1} there are $13k\lceil \log n \rceil$ points sampled from D_t uniformly at random, the probability that x remains to be angry is no more that the probability that all the q points haven't be selected, that is,

$$\Pr[\mathrm{Angry}(x)] \le (1 - \frac{q}{|D_t|})^{13k\lceil \log n \rceil} \le \exp\left(-\frac{q}{|D_t|} 13k\lceil \log n \rceil\right).$$

The expectation of the number of the angry points in D_t corresponding to c_x is at most

$$\sum_{q=1}^{+\infty} \exp\left(-\frac{q}{|D_t|} 13k\lceil \log n \rceil\right) \le \int_{x=0}^{+\infty} \exp\left(-\frac{x}{|D_t|} 13k\lceil \log n \rceil\right) dx \le \frac{|D_t|}{13k\lceil \log n \rceil}.$$

Since there are k centers in the optimal solution, the fraction of angry points in $|D_t|$ is $1/(13\lceil \log n \rceil)$.

For any happy point $x \in D_t$, there exists $c \in A_{t+1}$ such that $\mathrm{dist}(c, c_x) \le \mathrm{dist}(x, c_x)$. Hence, it holds that

$$\mathrm{dist}(x, A) \le \mathrm{dist}(x, A_{t+1}) \le \mathrm{dist}(x, c) \le \mathrm{dist}(x, c_x) + \mathrm{dist}(c_x, c) \le 2\,\mathrm{dist}(x, c_x).$$

For the angry points, sort all the points $x \in D_t$ from small to large by the key $\mathrm{dist}(x, \tau_{t+1}(x))$. Let each angry point grabs the first ungrabbed happy point and assume that s_t is an ungrabbed happy point. For any unhappy point $x \in D_t$, there is a happy point $y \in D_t$ preceding s_t in the sequence grabbed by x, since otherwise s_t is unhappy or grabbed and there comes a contradiction. Let G_{t+1} be

$$G_{t+1} := \{x \in D_t : \mathrm{dist}(x, \tau_{t+1}(x)) \le \mathrm{dist}(s_t, \tau_{t+1}(s_t))\}.$$

It can be concluded that for any unhappy point $x \in G_{t+1}$, there exists a unique happy point $y \in G_{t+1}$ such that $\mathrm{dist}(x, \tau_{t+1}(x)) \le \mathrm{dist}(y, \tau_{t+1}(y))$. Hence,

$$\sum_{x \in G_{t+1}, \mathrm{Angry}(x)} \mathrm{dist}^z(x, \tau_{t+1}(x)) \le \sum_{x \in G_{t+1}, \mathrm{Happy}(x)} \mathrm{dist}^z(x, \tau_{t+1}(x)).$$

For all the points in G_{t+1}, it holds that

$$\begin{aligned}
\mathrm{cost}(G_{t+1}, A) = \sum_{x\in G_{t+1}} \mathrm{dist}^z(x, A) &\le \sum_{x\in G_{t+1}} \mathrm{dist}^z(x, A_{t+1}) \\
&= \sum_{x\in G_{t+1}, \mathrm{Angry}(x)} \mathrm{dist}^z(x, A_{t+1}) + \sum_{x\in G_{t+1}, \mathrm{Happy}(x)} \mathrm{dist}^z(x, A_{t+1}) \\
&\le \sum_{x\in G_{t+1}, \mathrm{Angry}(x)} \mathrm{dist}^z(x, \tau_{t+1}(x)) + \sum_{x\in G_{t+1}, \mathrm{Happy}(x)} \mathrm{dist}^z(x, A_{t+1}) \\
&\le \sum_{x\in G_{t+1}, \mathrm{Happy}(x)} \mathrm{dist}^z(x, \tau_{t+1}(x)) + \sum_{x\in G_{t+1}, \mathrm{Happy}(x)} \mathrm{dist}^z(x, A_{t+1}) \\
&\le 2c_\tau^z \sum_{x\in G_{t+1}, \mathrm{Happy}(x)} \mathrm{dist}^z(x, A_{t+1}(x)) \\
&\le O(2^z c_\tau^z) \sum_{x\in G_{t+1}, \mathrm{Happy}(x)} \mathrm{dist}^z(x, c_x) \\
&\le O(2^z c_\tau^z) \sum_{x\in G_{t+1}} \mathrm{dist}^z(x, c_x).
\end{aligned}$$

The probability that s_t is ungrabbed happy point is the fraction of ungrabbed happy points in D_t, which is $2/(13\lceil \log n\rceil)$. The probability that all the s_t in the $3\lceil \log n\rceil$ iterations are ungrabbed and happy is at least $1 - \frac{2}{13\lceil \log n\rceil} \cdot 3\lceil \log n\rceil = 7/13$.

Due to the definition of G_t and D_t, $\cup_t G_t$ contains all the points deleted during the loop (Line 4-10). Besides, Since all the remaining points are added into A, they do not contribute to the cost. With a probability $7/13 - n^{-0.12} > 1/2$, it holds that

$$\begin{aligned}
\mathrm{cost}(D, A) &= \sum_{x\in D} \mathrm{dist}^z(x, A) \\
&= \sum_t \sum_{G_t} \mathrm{dist}^z(x, A) + \sum_{x\notin \cup_t G_t} \mathrm{dist}^z(x, A) \\
&\le O(2^z c_\tau^z) \sum_t \sum_{G_t} d(x, c_x) + 0 \\
&\le O(2^z c_\tau^z)\, \mathrm{OPT}\,.
\end{aligned}$$

In Line 12, Algorithm 1 repeats the processing in Line 3-11 for three times and union all the set A to boost the success probability to $5/6$. As a consequence, with probability at least $5/6$, the output A is an $(O(k\log^2 n), O(2^z c_\tau^z))$-bicriteria approximate solution. The proof of our complexity claim has been given in Section 3.1.

B. Further Proof Details for Coreset Construction Based on Biapproximate Solution

This section gives a detailed proof of Lemma 3.8. We restate this lemma as below.

Lemma B.1. *Let*

$$t = \tilde{O}\left(2^{O(z)} \cdot m \cdot (d + \log(n)) \cdot \max\left(\varepsilon^{-2}, \varepsilon^{-z}\right)\right)$$

in Algorithm 2. For a positive real $\varepsilon < 1/(4c_\tau^z)$, Algorithm 2 outputs an $O(c_\tau^z \beta\varepsilon)$-coreset of size $\tilde{O}\left(2^{O(z)} md\log(n)\max\left(\varepsilon^{-2}, \varepsilon^{-z}\right)\right)$ with probability at least $5/6$, using $\tilde{O}\left(2^{O(z)} c_\tau \sqrt{nmd}\max\left(\varepsilon^{-1}, \varepsilon^{-z/2}\right)\right)$ queries to O_τ, O_D, O_A, their inverses, and QRAM. Besides it uses $\mathrm{poly}(md\log n/\varepsilon^z)$ additional classical processing time.

This section first provides the detailed quantum implementation and the analysis of complexity in Appendix B.1, and then gives a proof that the output of Algorithm 2 (set Ω) is an $O(c_\tau^z\beta\varepsilon)$-coreset with probability at least $5/6$ in Appendix B.2.

B.1. Quantum Implementation and Complexity

This section shows the quantum implementation details with the complexity.

Proof for complexity. For Line 3, the algorithm estimates $|C_i|$ and $\mathrm{cost}_\tau(C_i, A)$ first. Constructing the oracle

$$\begin{aligned} U\colon\ & |s\rangle|0\rangle|0\rangle|0\rangle|0\rangle \xmapsto{O_\tau} |s\rangle|i\rangle|0\rangle|0\rangle|0\rangle \\ & \xmapsto{O_D,O_A} |s\rangle|i\rangle|x_s\rangle|a_i\rangle|0\rangle \\ & \mapsto |s\rangle|i\rangle|x_s\rangle|a_i\rangle|\mathrm{dist}^z(x_s,a_i)\rangle \\ & \xmapsto{O_A^{-1},O_D^{-1}} |s\rangle|i\rangle|0\rangle|0\rangle|\mathrm{dist}^z(x_s,a_i)\rangle \end{aligned}$$

and applying Theorem 4.2 yields the needed values $\mathrm{cost}_\tau(C_i, A)\ \forall i \in [m]$. Since $\mathrm{dist}^z(x_s, a_i) \le \mathrm{cost}_\tau(D, A) \le c_\tau^z\,\mathrm{OPT}$, the calculation uses no more than $\tilde{O}(z\log(c_\tau)\sqrt{nm}/\varepsilon)$ queries to U and additional time, under a fair assumption that $\mathrm{OPT} = \mathrm{poly}(n)$. The same technique works for $|C_i|$. Then the algorithm computes Δ_{C_i} in a classical manner. The implementation of Line 5, Line 7, and Line 8 is similar. These calculation uses in total no more than $\tilde{O}(z\log(c_\tau)\sqrt{nm}/\varepsilon)$ queries to U_R, U_G, O_τ, O_D, and O_A. Besides it uses $\mathrm{poly}\,(mz\log(1/\varepsilon))$ classical processing time.

The construction of the ring unitary U_R in Line 4 is

$$\begin{aligned} U_R\colon\ & |s\rangle|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle \\ & \xmapsto{O_\tau} |s\rangle|i\rangle|0\rangle|0\rangle|0\rangle|0\rangle \\ & \xmapsto{O_\Delta,O_D,O_A} |s\rangle|i\rangle|\Delta_{C_i}\rangle|x_s\rangle|a_i\rangle|0\rangle \\ & \mapsto |s\rangle|i\rangle|\Delta_{C_i}\rangle|x_s\rangle|a_i\rangle|j\rangle \\ & \xmapsto{O_A^{-1},O_D^{-1},O_\Delta^{-1}} |s\rangle|i\rangle|0\rangle|0\rangle|0\rangle|j\rangle \end{aligned}$$

where $j = \lfloor \log\left(\mathrm{dist}^z(x_s, a_i)/\Delta_{C_i}\right)\rfloor$, and $O_\Delta\colon\ |i\rangle|0\rangle \to |i\rangle|\Delta_{C_i}\rangle\ \forall i \in [m]$ is constructed by storing Δ_{C_i} in QRAM in Line 3. One query to U_R needs constant queries to O_D, O_A, O_τ, and QRAM. The same technique works for the construction of U_G and the complexity is also the same up to a constant factor.

For Line 9, the algorithm first construct the below unitary U for each well-sturctured G.

$$\begin{aligned} U\colon\ & |s\rangle|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle \\ & \xmapsto{U_G,O_\tau} \xmapsto{O_\Delta} \mapsto |s\rangle|j\rangle|b\rangle|i\rangle|\Delta_{C_i}\rangle|I(x_s \in G)\rangle|p_s\rangle \\ & \mapsto \xmapsto{O_\Delta} \xmapsto{O_\tau^{-1},U_G^{-1}} |s\rangle|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle|p_s\rangle \end{aligned}$$

where $I(x_s \in G)$ is the indicator for whether $x_s \in G$ and $p_i = I(x_s \in G)\Delta_{C_i}$ is proportional to $\Pr[x_s]$. Then one application to Lemma 2.5 yields the sample Ω using $O(\sqrt{nt})$ queries to the above unitary U. Reweighting can be completed in a classical manner since U is of small size. For Line 10 the algorithm uses the same technique. The sampling process in total needs $\tilde{O}(\sqrt{nt})$ queries to U_G, O_A, O_τ, and QRAM. It uses additional $\mathrm{poly}(m\log n) + O(t)\cdot d\,\mathrm{polylog}(mn)$ for computing the weight classically.

Let $t = \tilde{O}\left(2^{O(z)}\cdot m\cdot(d+\log(n))\cdot\max\left(\varepsilon^{-2},\varepsilon^{-z}\right)\right)$ and sum up all the time cost. Algorithm 2 uses $\tilde{O}\left(2^{O(z)}c_\tau\sqrt{nmd}\max\left(\varepsilon^{-1},\varepsilon^{-z/2}\right)\right)$ queries to O_τ, O_D, O_A and QRAM, and $\mathrm{poly}(md\log n/\varepsilon^z)$ additional classical processing time.

Similar to Algorithm 1, each subroutine used in Algorithm 2 suffers only a $\log(1/\delta)$ factor to reach success probability at least $1-\delta$ and each subroutine is applied no more than $\mathrm{poly}(nmz)$ times, so it is enough to set the failure probability as $\delta = O(1/\mathrm{poly}(nmz))$ for each subroutine. This cause only a $\mathrm{polylog}(nmz)$ factor on time consume and it is adsorbed by the $\tilde{O}$ notation. □

B.2. Correctness

This section gives a rigorous proof that set Ω, the output of Algorithm 2, is an $O(c_\tau^z\beta\varepsilon)$-coreset with probability at least $5/6$. This proof follows the idea of Cohen-Addad et al. (2021).

The dataset D can be seen as a partition of the following three kinds of points:

- the union of the inner rings R_I, the cheapest groups $G_{j,\min}$ with $j \in [z\log(4\varepsilon/z), z\log(4z/\varepsilon)]$ and $G_{O,\min}$
- the well-structured groups $G_{j,b}$ with $j \in [z\log(\varepsilon/4z), z\log(4z/\varepsilon)]$ and $b = 1, \ldots, \max$
- the outer rings $G_{O,b}$ with $b = 1, \ldots, \max$.

Algorithm 2 deals with this three kinds of points separately and so does the following proof. The following proof shows that, let $\varepsilon \le 1/(4c_\tau^z)$ and

$$t = \tilde{O}\left(2^{O(z)} \cdot m \cdot (d + \log(n)) \cdot \max\left(\varepsilon^{-2}, \varepsilon^{-z}\right)\right)$$

in Algorithm 2, this algorithm has the following three properties, which are stated formally and proved later.

- **Lemma B.4** Let $B := R_I \cup G_{j,\min}$ be the set of the first kind of points. For any $S \in (\mathbb{R}^d)^m$ it holds that

$$|\operatorname{cost}(B, S) - \operatorname{cost}(A, S)| \le 8\varepsilon\left(\operatorname{cost}(D, S) + \operatorname{cost}_\tau(D, A)\right).$$

- **Lemma B.7** It holds with probability at least $1/12$ that for any well-structured group $G = G_{j,b}$ and the corresponding sample $\Omega = \Omega_{j,b}$, and for any $S \in (\mathbb{R}^d)^m$,

$$|\operatorname{cost}(G, S) - \operatorname{cost}(\Omega, S)| = O(c_\tau^z\varepsilon)\left(\operatorname{cost}(G, S) + \operatorname{cost}(G, A)\right).$$

- **Lemma B.12** It holds with probability at least $1/12$ that, for any outer group $G = G_{O,b}$ and the corresponding sample $\Omega_{O,b}$, and for any $S \in (\mathbb{R}^d)^m$,

$$|\operatorname{cost}(G, S) - \operatorname{cost}(\Omega, S)| \le \frac{2c_\tau^z\varepsilon}{z\log(z/\varepsilon)}(\operatorname{cost}(D, S) + \operatorname{cost}(D, A)).$$

Note that there are only $O(z\log(z/\varepsilon))$ outer groups $G_{O,b}$. Combining the three properties directly yields the proof for correctness.

Proof for correctness.

$$|\operatorname{cost}(D, S) - \operatorname{cost}(\Omega, S)| \le O(c_\tau^z\varepsilon)\left(\operatorname{cost}(D, S) + \operatorname{cost}(D, A)\right) \le O(c_\tau^z\beta\varepsilon)\operatorname{cost}(D, S).$$

Therefore, the output of Algorithm 2 $\Omega = A \cup \Omega_{j,b} \cup \Omega_{O,b}$ is an $O(c_\tau^z\beta\varepsilon)$-coreset. □

The two lemmas introduced as follows are important tools for the proof.

Lemma B.2 (Triangle Inequality of Powers)**.** *Let a, b, and c be three arbitrary sets of points $\mathbb{R}^d$. For any $z \in \mathbb{Z}_+$ and any $\varepsilon > 0$, it holds that*

$$\operatorname{dist}^z(a, b) \le (1+\varepsilon)^{z-1}\operatorname{dist}^z(a, c) + \left(\frac{1+\varepsilon}{\varepsilon}\right)^{z-1}\operatorname{dist}^z(b, c)$$

$$|\operatorname{dist}^z(a, S) - \operatorname{dist}^z(b, S)| \le \varepsilon d^z(a, S) + \left(\frac{2z+\varepsilon}{\varepsilon}\right)^{z-1}\operatorname{dist}^z(a, b).$$

Lemma B.3 (Rephrased from Definition 1 and Lemma 17 of Cohen-Addad et al. 2021)**.** *There is a set $\mathbb{C}$ of size $n \cdot (z/\varepsilon)^{O(d)}$ such that, for any solution $S \in (\mathbb{R}^d)^m$ there exists $\tilde{S} \in \mathbb{C}^m$ which ensures that for any point $x \in D$ with either $\operatorname{cost}(x, S) \le (8z/\varepsilon)^z \operatorname{cost}_\tau(x, A)$ or $\operatorname{cost}(x, \tilde{S}) \le (8z/\varepsilon)^z \operatorname{cost}_\tau(x, A)$, it holds that*

$$|\operatorname{cost}(x, S) - \operatorname{cost}(x, \tilde{S})| \le \varepsilon\left(\operatorname{cost}(x, S) + \operatorname{cost}_\tau(x, A)\right).$$

Such a set $\mathbb{C}$ is called an A-approximate centroid set *for (m, z)-clustering on data set D.*

Proof of Lemma B.3. For any $x \in \mathbb{R}^d$ and $r \geq 0$, let $B(x,r) := \{y \in \mathbb{R}^d \mid \operatorname{dist}(x,y) \leq r\}$ be the *ball* around x with radius r. Note that Euclidean space $\mathbb{R}^d$ has *doubling dimension* $O(d)$, which means in this metric space any ball of radius $2r$ can be covered by $2^{O(d)}$ balls of radius r. For any $V \subset \mathbb{R}^d$, a γ-*net* of V is a set of points $X \subset V$ such that for any $v \in V$ there exists $x \in X$ such that $\operatorname{dist}(x,v) \leq \gamma$ and for any $x, y \in X$ it holds that $\operatorname{dist}(x,y) > \gamma$. In $\mathbb{R}^d$, a point set $V \subset \mathbb{R}^d$ with diameter D has a γ-net with size $2^{O(d\log(D/\gamma))}$ (Gupta et al., 2003).

Let data points $x_1, \ldots, x_n$ be in order with non-descreasing value of $\operatorname{dist}(x, a_{\tau(x)})$. Let N_i be an $\varepsilon \cdot \operatorname{dist}(x, a_{\tau(x)})/(4z)$-net of $B\left(x_i, 10z \operatorname{dist}(x_i, a_{\tau(x_i)})/\varepsilon\right) \setminus \cup_{j<i} B\left(x_j, 10z \operatorname{dist}(x_j, a_{\tau(x_j)})/\varepsilon\right)$. Let $s_f \in \mathbb{R}^d$ be a point such that $s_f \notin B\left(x_i, 10z \operatorname{dist}(x_i, a_{\tau(x_i)})/\varepsilon\right)\ \forall i \in [n]$. Let $N := \cup_{x_i \in D} N_i \cup \{s_f\}$.

The size of N is bound by $n \cdot (z/\varepsilon)^{O(d)}$:

$$|N| \leq n \cdot 2^{O(d\log(z/\varepsilon)^2)} = n \cdot \left(\frac{z}{\varepsilon}\right)^{O(d)}.$$

N is an A-approximated centroid set. For any solution $S \in (\mathbb{R}^d)^m$, let $\tilde{S} \in N^m$ be constructed by the following method. For each point $s \in S$, let i be the smallest index such that $s \in B\left(x_i, 10z \operatorname{dist}(x_i, a_{\tau(x_i)})/\varepsilon\right)$. The corresponding N_i is non-empty because otherwise there exists x_j such that $j < i$ and $s \in B(x_j, 10z \operatorname{dist}(x_j, a_{\tau(x_j)})/\varepsilon)$, and thus i is not the smallest index. Let $\tilde{s}$ be the closest point to s in N_i. If such an index i does not exist, let $\tilde{s} = s_f$. Let $\tilde{S}$ be the set of all the $\tilde{s}$. $\tilde{S}$ has the property defined in Lemma B.3.

Let $x \in D$ satisfies $\operatorname{cost}(x,S) \leq (10z/\varepsilon)^z \operatorname{cost}_\tau(x,A)$. Let s be the nearest neighbor of x in S and consider the corresponding index i and $\tilde{s}$. It holds that $\operatorname{dist}(x, a_{\tau(x)}) \geq \operatorname{dist}(x_i, a_{\tau(x_i)})$ since $s \in B\left(x, 10z \operatorname{dist}(x, a_{\tau(x)})/\varepsilon\right)$. By the definition of $\tilde{s}$ it holds that $\operatorname{dist}(s,\tilde{s}) \leq \varepsilon \operatorname{dist}(x_i, a_{\tau(x_i)})/(4z) \leq (\varepsilon/4z) \operatorname{dist}_\tau(x,A)$. As a consequence,

$$\begin{aligned}\operatorname{cost}(x,\tilde{S}) &\leq \operatorname{cost}(x,\tilde{s}) \leq (1+\varepsilon)\operatorname{cost}(p,s) + (1+z/\varepsilon)^{z-1}\operatorname{cost}(s,\tilde{s}) \\ &\leq (1+\varepsilon)\operatorname{cost}(x,S) + \varepsilon \operatorname{cost}_\tau(x,A).\end{aligned}$$

On the other hand, let $x \in D$ satisfies $\operatorname{cost}(x,\tilde{S}) \leq (10z/\varepsilon)^z \operatorname{cost}_\tau(x,A)$. Let $\tilde{s}$ be the nearest neighbor of x in $\tilde{S}$ and consider the corresponding s and index i. If the index of x is smaller than i it can be implied that $\tilde{s} \notin N_i$ because of the definition of N_i and $\tilde{s} \in B\left(x, 10z \operatorname{dist}(x, a_{\tau(x)})/\varepsilon\right)$. As a consequence, $\operatorname{dist}(x, a_{\tau(x)}) \geq \operatorname{dist}(x_i, a_{\tau(x_i)})$. It holds that

$$\begin{aligned}\operatorname{cost}(x,S) &\leq \operatorname{cost}(x,s) \leq (1+\varepsilon)\operatorname{cost}(x,\tilde{s}) + (1+2z/\varepsilon)^{z-1}\operatorname{cost}(s,\tilde{s}) \\ &\leq (1+\varepsilon)\operatorname{cost}(x,\tilde{S}) + \varepsilon \operatorname{cost}_\tau(x,A).\end{aligned}$$

For any $x \in D$ with $\operatorname{cost}(x,S) \leq (8z/\varepsilon)^z \operatorname{cost}_\tau(x,A)$, it holds that

$$\operatorname{cost}(x,\tilde{S}) \leq (1+\varepsilon)\operatorname{cost}(x,S) + \varepsilon \operatorname{cost}_\tau(x,A) \leq (10z/\varepsilon)^z \operatorname{cost}_\tau(x,A).$$

Thus $\operatorname{cost}(x,S) \leq (1+\varepsilon)\operatorname{cost}(x,\tilde{S}) + \varepsilon \operatorname{cost}_\tau(x,A)$. Hence

$$|\operatorname{cost}(x,S) - \operatorname{cost}(x,\tilde{S})| \leq \varepsilon\left(\operatorname{cost}(x,S) + \operatorname{cost}_\tau(x,A)\right).$$

The same inequality holds for any $x \in D$ with $\operatorname{cost}(x.\tilde{S}) \leq (8z/\varepsilon)^z \operatorname{cost}_\tau(x,A)$. □

The first kind of points Let $B := R_I \cup G_{j,\min} \cup G^O_{\min}$ be the set of the first kind of points. There holds the following lemma:

Lemma B.4. *For any solution S, $|S| \leq m$ and $\varepsilon < 1/2$ it holds that*

$$|\operatorname{cost}(B,S) - \operatorname{cost}(A,S)| \leq 8\varepsilon\left(\operatorname{cost}(D,S) + \operatorname{cost}_\tau(D,A)\right).$$

We use the following two lemmas to prove Lemma B.4:

Lemma B.5. *For any solution S, $|S| \leq m$, any $i \in [m]$, and $\varepsilon < 1/2$,*

$$|\operatorname{cost}(R_I(C_i),S) - |R_I(C_i)| \cdot \operatorname{cost}(a_i,S)| \leq \varepsilon \operatorname{cost}(R_I(C_i),S) + 2\varepsilon \operatorname{cost}_\tau(C_i,A).$$

Lemma B.6. *For any solution S, $|S| \le m$ and any cheapest group G,*

$$|\operatorname{cost}(G,S) - \sum_{i=1}^{m} |C_i \cap G| \operatorname{cost}(a_i,S)| \le \varepsilon \operatorname{cost}(R_j,S) + \varepsilon \operatorname{cost}_\tau(R_j,A)$$

if $G = G_{j,\min}$ for some $j \neq O$. And if $G = G^O_{\min}$ there is

$$|\operatorname{cost}(G,S) - \sum_{i=1}^{m} |C_i \cap G| \operatorname{cost}(a_i,S)| \le \varepsilon \operatorname{cost}(D,S) + \varepsilon \operatorname{cost}_\tau(D,A).$$

Proof of Lemma B.5. Fix i. Using Lemma B.2, it holds that

$$|\operatorname{cost}(x,S) - \operatorname{cost}(a_i,S)| = |\operatorname{dist}^z(x,S) - \operatorname{dist}^z(a_i,S)| \le \varepsilon \operatorname{dist}^z(x,S) + (1+\frac{2z}{\varepsilon})^{z-1} \operatorname{dist}^z(a_i,x).$$

For any $x \in R_I(C_i)$ and $\varepsilon < 1/2$, there is

$$(1+2z/\varepsilon)^{z-1} \operatorname{dist}^z(a_i,x) \le (1+2z/\varepsilon)^{z-1} \operatorname{cost}_\tau(x,A) \le (\frac{3z}{\varepsilon})^{z-1}(\frac{\varepsilon}{4z})^z \Delta_{C_i} \le \varepsilon \frac{\operatorname{cost}_\tau(C_i,A)}{(1-\varepsilon)|C_i|} \le 2\varepsilon \frac{\operatorname{cost}_\tau(C_i,A)}{|R_I(C_i)|}$$

Combining the two inequalities above and summing the result over all the points in R_I yields

$$\begin{aligned} |\operatorname{cost}(R_I(C_i),S) - |R_I(C_i)| \cdot \operatorname{cost}(a_i,S)| &\le \sum_{x \in R_I(C_i)} |\operatorname{cost}(x,S) - \operatorname{cost}(a_i,S)| \\ &\le \varepsilon \operatorname{cost}(R_I(C_i),S) + 2\varepsilon \operatorname{cost}_\tau(C_i,A). \end{aligned}$$

□

Proof of Lemma B.6. Using Lemma B.2, it holds that

$$|\operatorname{cost}(x,S) - \operatorname{cost}(a_{\tau(x)},S)| = |\operatorname{dist}^z(x,S) - \operatorname{dist}^z(a_{\tau(x)},S)| \le \varepsilon \operatorname{dist}^z(x,S) + \left(1+\frac{2z}{\varepsilon}\right)^{z-1} \operatorname{dist}^z(a_{\tau(x)},x).$$

The sum over G tells

$$\begin{aligned} |\operatorname{cost}(G,S) - \sum_{i=1}^{m} |C_i \cap G| \operatorname{cost}(a_i,S)| &= |\sum_{x \in G} \operatorname{cost}(x,S) - \sum_{x \in G} \operatorname{cost}(a_{\tau(x)},S)| \\ &\le \sum_{x \in G} \left(\varepsilon \operatorname{dist}^z(x,S) + (1+\frac{2z}{\varepsilon})^{z-1} \operatorname{dist}^z(a_{\tau(x)},x) \right) \\ &\le \varepsilon \operatorname{cost}(G,S) + (\frac{3z}{\varepsilon})^{z-1} \operatorname{cost}_\tau(G,A). \end{aligned}$$

If $G = G_{j,\min}$ for some j, it holds that $\operatorname{cost}_\tau(G,A) \le (\varepsilon/4z)^z \cdot \operatorname{cost}_\tau(R_j,A)$. Else $G = G^O_{\min}$ and $\operatorname{cost}_\tau(G,A) \le (\varepsilon/4z)^z \cdot \operatorname{cost}_\tau(R_O,A) \le (\varepsilon/4z)^z \cdot \operatorname{cost}_\tau(D,A)$. In both cases the lemma holds. □

Combining Lemma B.5 and Lemma B.6 straightforwardly gives Lemma B.4. As is talked about in Section 3.2, due to the particularity of quantum computing, it is costly to compute the exact value $|R_{i,I}| + |C_i \cap (\cup_{j \neq I} G_{j,\min})|$. What the algorithm uses is ε-estimations $\tilde{r}_i$ such that

$$|\tilde{r}_i - (|R_I(C_i)| + |C_i \cap (\cup_{j \notin \{I,O\}} G_{j,\min})| + |C_i \cap G^O_{\min}|) \le \varepsilon(|R_I(C_i)| + |C_i \cap (\cup_{j \notin \{I,O\}} G_{j,\min})| + |C_i \cap G^O_{\min}|)$$

for any $i \in [m]$.

Proof of Lemma B.4.

$$
\begin{aligned}
&|\operatorname{cost}(B,S)-\operatorname{cost}(A,S)|\\
=&|\operatorname{cost}(B,S)-\sum_{i=1}^{m}\tilde{r}_i\operatorname{cost}(a_i,S)|\\
=&|(1\pm\varepsilon)\operatorname{cost}(B,S)\mp\varepsilon\operatorname{cost}(B,S)-(1\pm\varepsilon)\sum_{i=1}^{m}(|R_I(C_i)|+|C_i\cap(\cup_j G_{j,\min})|+|C_i\cap G^O_{\min}|)\operatorname{cost}(a_i,S)|\\
\le&(1+\varepsilon)\left(\sum_{i=1}^{m}|(\operatorname{cost}(R_I(C_i),S)-|R_I(C_i)|\operatorname{cost}(a_i,S)|)+\sum_{G}(|\operatorname{cost}(C\cap G,S)-\sum_{i=1}^{m}|C_i\cap G|\operatorname{cost}(a_i,S)|)\right)\\
&+\varepsilon\operatorname{cost}(B,S)\\
\le&(1+\varepsilon)\varepsilon\left(\sum_{i=1}^{m}(\operatorname{cost}(R_I(C_i),S)+2\operatorname{cost}_\tau(C_i,A))+\sum_{j}(\operatorname{cost}(R_j,S)+\operatorname{cost}_\tau(R_j,A)))+\operatorname{cost}(D,S)+\operatorname{cost}_\tau(D,A)\right)\\
&+\varepsilon\operatorname{cost}(B,S)\\
\le&(1+\varepsilon)\varepsilon(3\operatorname{cost}(D,S)+4\operatorname{cost}_\tau(D,A))+\operatorname{cost}(D,S)\\
\le&8\varepsilon\left(\operatorname{cost}(D,S)+\operatorname{cost}_\tau(D,A)\right).
\end{aligned}
$$

□

Well-structured groups For well-structured groups, the following lemma holds:

Lemma B.7. *Let*

$$t=\tilde{O}\left(2^{O(z)}\cdot m\cdot(d+\log(n))\cdot\max\left(\varepsilon^{-2},\varepsilon^{-z}\right)\right)$$

in Algorithm 2 Line 9. For $\varepsilon<1/(4c_\tau^z)$, it holds with probability at least $1/12$ that for any well-structured group $G=G_{j,b}$ and the corresponding sample $\Omega=\Omega_{j,b}$, and for any $S\in(\mathbb{R}^d)^m$,

$$|\operatorname{cost}(G,S)-\operatorname{cost}(\Omega,S)|=O(c_\tau^z\varepsilon)\left(\operatorname{cost}(G,S)+\operatorname{cost}(G,A)\right).$$

Fix a well-structured group G and for convenience, in the following for any set $C\subset\mathbb{R}^d$ we write $G\cap C$ simply as C. Due to the definition of G, for every cluster C_i, the following properties hold:

- $\forall x,y\in C_i$, $\operatorname{cost}_\tau(x,A)\le 2\operatorname{cost}_\tau(y,A)$,
- $\operatorname{cost}_\tau(G,A)/(2m)\le\operatorname{cost}_\tau(C_i,A)$,
- $\forall x\in C_i$, $\operatorname{cost}_\tau(C_i,A)/(2|C_i|)\le\operatorname{cost}_\tau(x,A)\le(2\operatorname{cost}_\tau(C_i,A))/|C_i|$.

Recall that Ω is an i.i.d sample of size t and in each round a point $x\in C_i\cap G$ is sampled with probability

$$\Pr[x]=\frac{\operatorname{cost}_\tau(C_i,A)}{|C_i|\operatorname{cost}_\tau(G,A)}$$

and for any $x\in\Omega$,

$$w(x)=\frac{|C_i|\operatorname{cost}_\tau(G,A)}{|\Omega|\operatorname{cost}_\tau(C_i,A)}.$$

To be precisely, the values such as $|C_i|$ and $\operatorname{cost}_\tau(C_i,A)$ are ε-estimations in practice, instead of the exact values shown above. The method to deal with such problems is the same as in the proof of Lemma B.4. For convenience we do not repeat similar proof and use the exact values directly.

It can be seen that given a sample large enough, $|C_i|$ can be well approximated for every $i\in[m]$ (Lemma B.8). The proof of Lemma B.8 is given later.

Lemma B.8. *Define event $\mathcal{E}$ to be for any $i \in [m]$,*

$$\sum_{x \in C_i \cap \Omega} \frac{|C_i| \operatorname{cost}_\tau(G, A)}{|\Omega| \operatorname{cost}_\tau(C_i, A)} = (1 \pm \varepsilon)|C_i|.$$

With probability at least $1 - 2m \cdot \exp\left(-(\varepsilon^2 t)/(6m)\right)$, event $\mathcal{E}$ holds.

For any solution S, let

$$I_{l,S} := \{x \in G \mid 2^l \operatorname{cost}_\tau(x, A) \le \operatorname{cost}(x, S) \le 2^{l+1} \operatorname{cost}_\tau(x, A)\},$$

and let these $\{I_{l,S}\}$ be divided into three parts: tiny ranges, with $l \le \log(\varepsilon/2)$; interesting ranges, with $\log(\varepsilon/2) \le l \le z\log(4z/\varepsilon)$; and huge ranges, with $l \ge z\log(4z/\varepsilon)$. For the three types of $I_{l,S}$, there exist the following lemmas, respectively:

Lemma B.9. *Let $I_{\text{tiny},S} := \cup_{l \le \log(\varepsilon/2)} I_{l,S}$. For any solution S, it holds that*

$$\max\left(\operatorname{cost}(I_{\text{tiny},S}, S), \operatorname{cost}(I_{\text{tiny},S} \cap \Omega, S)\right) \le \varepsilon \operatorname{cost}_\tau(G, A).$$

Lemma B.10. *Condition on event $\mathcal{E}$, it holds that*

$$|\operatorname{cost}(C_i, S) - \operatorname{cost}(C_i \cap \Omega, S)| \le O(\varepsilon) \operatorname{cost}(C_i, S)$$

for any solution S, and any cluster C_i such that there exists a huge range $I_{l,S}$ with $I_{l,S} \cap C_i \neq \varnothing$.

Lemma B.11. *Let $L_S := \{C_i \mid \forall x \in C_i, \operatorname{cost}(x, S) \le (4z/\varepsilon)^z \operatorname{cost}(x, A)\}$. Let $\mathbb{C}$ be an A-approximate centroid set of size $n \cdot (z/\varepsilon)^{O(d)}$ as defined in Lemma B.3. With probability*

$$1 - \exp\left(m\log(n) + O(md\log(z/\varepsilon)) - 2^{O(z\log z)} \cdot \frac{\min(\varepsilon^2, \varepsilon^z)}{\log^2(1/\varepsilon)} \cdot t\right)$$

and together with event $\mathcal{E}$, it holds that for any solution $S \in \mathbb{C}^m$

$$|\operatorname{cost}(L_S, S) - \operatorname{cost}(L_S \cap \Omega, S)| \le \varepsilon(\operatorname{cost}_\tau(G, A) + \operatorname{cost}(G, S)).$$

Using Lemma B.10 and Lemma B.11, Lemma B.7 can be proved.

Proof of Lemma B.7. Let G be an arbitrary well-structured group. Let S be an arbitrary solution and let $\tilde{S} \in \mathbb{C}^k$ approximate S. Denote

$$H_S := \{x \in G \mid \exists i, x \in C_i \text{ and } \exists l > z\log(8z/\varepsilon), C_i \cap I_{l,S} \neq \varnothing\}$$
$$H_{\tilde{S}} := \{x \in G \mid \exists i, x \in C_i \text{ and } \exists l > z\log(4z/\varepsilon), C_i \cap I_{l,\tilde{S}} \neq \varnothing\} \setminus H_S.$$

And denote $L_{\tilde{S}}$ as in Lemma B.11. It holds that H_S, $H_{\tilde{S}}$, and $L_{\tilde{S}}$ form a partition of G. On the one hand, $H_S \cup H_{\tilde{S}} \cup L_{\tilde{S}} = G$. On the other hand, $H_S \cap H_{tildeS} = \varnothing$, $H_{\tilde{S}} \cap L_{\tilde{S}} = \varnothing$, and $L_{\tilde{S}} \cap H_S = \varnothing$ since $\forall x \in L_{\tilde{S}} \operatorname{cost}(x, \tilde{S}) \le (4z/\varepsilon)^z \operatorname{cost}_\tau(x, A)$, thus $\operatorname{cost}(x, S) \le (1+\varepsilon)\operatorname{cost}(x, \tilde{S}) + \varepsilon \operatorname{cost}_\tau(x, A) \le (8z/\varepsilon)^z \operatorname{cost}_\tau(x, A)$.

By this partition and the property of $\tilde{S}$, it holds that

$$\begin{aligned}
&|\operatorname{cost}(G, S) - \operatorname{cost}(\Omega, S)| \\
=&|\sum_{x \in H_S} \operatorname{cost}(x, S) - \sum_{x \in H_S \cap \Omega} w(x) \operatorname{cost}(x, S)| + |\sum_{x \in G \setminus H_S} \operatorname{cost}(x, S) - \sum_{x \in (G \setminus H_S) \cap \Omega} w(x) \operatorname{cost}(x, S)| \\
\le&|\sum_{x \in H_S} \operatorname{cost}(x, S) - \sum_{x \in H_S \cap \Omega} w(x) \operatorname{cost}(x, S)| + |\sum_{x \in G \setminus H_S} \operatorname{cost}(x, \tilde{S}) - \sum_{x \in (G \setminus H_S) \cap \Omega} w(x) \operatorname{cost}(x, \tilde{S})| \\
&+ \varepsilon\left(\operatorname{cost}(G, S) + \operatorname{cost}_\tau(G, A) + \operatorname{cost}(\Omega, S) + \operatorname{cost}_\tau(\Omega, A)\right) \\
\le&|\sum_{x \in H_S} \operatorname{cost}(x, S) - \sum_{x \in H_S \cap \Omega} w(x) \operatorname{cost}(x, S)| + \varepsilon\left(\operatorname{cost}(G, S) + \operatorname{cost}_\tau(G, A) + \operatorname{cost}(\Omega, S) + \operatorname{cost}_\tau(\Omega, A)\right) \\
&+ |\sum_{x \in L_{\tilde{S}}} \operatorname{cost}(x, S) - \sum_{x \in L_{\tilde{S}} \cap \Omega} w(x) \operatorname{cost}(x, S)| + |\sum_{x \in H_{\tilde{S}}} \operatorname{cost}(x, S) - \sum_{x \in H_{\tilde{S}} \cap \Omega} w(x) \operatorname{cost}(x, S)| \\
\le&O(\varepsilon)\left(\operatorname{cost}(G, S) + \operatorname{cost}_\tau(G, A) + \operatorname{cost}(\Omega, S) + \operatorname{cost}_\tau(\Omega, A)\right) \\
\le&O(c_\tau^z \varepsilon)\left(\operatorname{cost}(G, S) + \operatorname{cost}(G, A) + \operatorname{cost}(\Omega, S) + \operatorname{cost}(\Omega, A)\right).
\end{aligned}$$

The last inequality uses Lemma B.10 and Lemma B.11.

Assume that $\varepsilon < 1/(4c_\tau^z)$. Let $S = A$, it holds that

$$\operatorname{cost}(\Omega, A) \le \operatorname{cost}(G, A) + |\operatorname{cost}(G, A) - \operatorname{cost}(\Omega, A)| \le O(1)\operatorname{cost}(G, A).$$

Similarly,

$$\operatorname{cost}(\Omega, S) \le \operatorname{cost}(G, S) + |\operatorname{cost}(G, S) - \operatorname{cost}(\Omega, S)| \le O(1)\left(\operatorname{cost}(G, S) + \operatorname{cost}(G, A)\right).$$

Hence it can be concluded that

$$|\operatorname{cost}(G, S) - \operatorname{cost}(\Omega, S)| = O(c_\tau^z \varepsilon)\left(\operatorname{cost}(G, S) + \operatorname{cost}(G, A)\right)$$

Using the union bound over event $\mathcal{E}$ and the probability of Lemma B.11 for all the well-structured groups G, the probability is

$$1 - z^2 \log^2(z/\varepsilon)\left(\exp\left(m\log(n) + O(md\log(z/\varepsilon)) - 2^{O(-z\log z)} \cdot \frac{\min(\varepsilon^2, \varepsilon^z)}{\log^2(1/\varepsilon)} \cdot t\right) - 2m \cdot \exp\left(-(\varepsilon^2 t)/(6m)\right)\right).$$

The probability can be bound by $1/12$ by setting the value of t as

$$t = \tilde{O}\left(2^{O(z)} \cdot m \cdot (d + \log(n)) \cdot \max\left(\varepsilon^{-2}, \varepsilon^{-z}\right)\right).$$

□

The proofs ofLemma B.8, Lemma B.9, Lemma B.10, and Lemma B.11 are shown as below, respectively. Lemma B.8 is used in the proof of Lemma B.10 and Lemma B.11, and Lemma B.9 is used in the proof of Lemma B.11.

Proof of Lemma B.8. Fix $i \in [m]$. Define $P_i(x)$ as the indicator of point $x \in \Omega$ being drawn from C_i, i.e., $P_i(x) = 1$ if $x \in C_i \cap \Omega$, and otherwise $P_i = 0$. The expectation of $P_i(x)$ has the following property:

$$\mathbb{E}(P_i(x)) = \sum_{x \in C_i} |\Omega| \Pr[x] = \sum_{x \in C_i} \frac{|\Omega| \operatorname{cost}_\tau(C_i, A)}{|C_i| \operatorname{cost}_\tau(G, A)} \ge \frac{|\Omega|}{2m}.$$

By Chernoff bounds, it holds that

$$\Pr[|\sum_{x \in \Omega} P_i(x) - \mathbb{E}(P_i(x))| \ge \varepsilon \mathbb{E}(P_i(x))] \le 2e^{-\varepsilon^2 \mathbb{E}(P_i(x))/3} \le 2e^{-(\varepsilon^2|\Omega|)/(6m)}.$$

The union bound over all the clusters derives that, with probability at least $1 - 2m \cdot \exp\left(-(\varepsilon^2 t)/(6m)\right)$, for any cluster C_i there is

$$|C_i \cap \Omega| = (1 \pm \varepsilon) \sum_{x \in C_i} \frac{|\Omega| \operatorname{cost}_\tau(C_i, A)}{|C_i| \operatorname{cost}_\tau(G, A)}$$

which implies

$$\sum_{x \in C_i \cap \Omega} \frac{|C_i| \operatorname{cost}_\tau(G, A)}{|\Omega| \operatorname{cost}_\tau(C_i, A)} = (1 \pm \varepsilon)|C_i|.$$

□

Proof of Lemma B.9.

$$\begin{aligned}
\operatorname{cost}(I_{\text{tiny},S}, S) &= \sum_{x \in I_{\text{tiny},S}} \operatorname{cost}(x, S) \le |I_{\text{tiny},S}| \frac{\varepsilon}{2} \operatorname{cost}_\tau(x, A) \le \varepsilon \operatorname{cost}_\tau(G, A) \\
\operatorname{cost}(I_{\text{tiny},S} \cap \Omega, S) &= \sum_{x \in I_{\text{tiny},S} \cap \Omega} w(x) \operatorname{cost}(x, S) \\
&= \sum_{x \in I_{\text{tiny},S} \cap \Omega} \frac{|C_i| \operatorname{cost}_\tau(G, A)}{t \operatorname{cost}_\tau(C_i, A)} \operatorname{cost}(x, S) \\
&\le |I_{,S} \cap \Omega| \frac{|C_i| \operatorname{cost}_\tau(G, A)}{|\Omega| \operatorname{cost}_\tau(C_i, A)} \cdot \frac{\varepsilon}{2} \cdot \frac{2 \operatorname{cost}_\tau(C_i, A)}{|C_i|} \\
&\le \varepsilon \operatorname{cost}_\tau(G, A).
\end{aligned}$$

□

Proof of Lemma B.10. For any $x \in C_i \cap \Omega$, $\mathrm{cost}(x, S)$ is bound. Let $y \in I_{l,S} \cap C_i$ with $I_{l,S}$ being a huge range. For any $x \in C_i$, there is

$$\mathrm{cost}(x, y) \le (\mathrm{dist}(x, A) + \mathrm{dist}(y, A))^z \le 3^z \,\mathrm{cost}_\tau(y, A) \le 3^z 2^{l - z\log(4z/\varepsilon)} \,\mathrm{cost}_\tau(y, A) \le \left(\frac{3\varepsilon}{4z}\right)^z \mathrm{cost}(y, S).$$

Using Lemma B.2, there is

$$\begin{aligned}\mathrm{cost}(y, S) &\le \left(1 + \frac{\varepsilon}{2z}\right)^{z-1} \mathrm{cost}(x, S) + \left(1 + \frac{2z}{\varepsilon}\right)^{z-1} \mathrm{cost}(x, y) \\ &\le (1 + \varepsilon)\,\mathrm{cost}(x, S) + \varepsilon\,\mathrm{cost}(y, S)\end{aligned}$$

which implies $\mathrm{cost}(x, S) \ge (1 - 2\varepsilon)\,\mathrm{cost}(y, S)$ and $\mathrm{cost}(y, S) \le (1 + 3\varepsilon)\,\mathrm{cost}(x, S)$ if $\varepsilon < 1/3$. Similarly $\mathrm{cost}(x, S) \le (1 + 2\varepsilon)\,\mathrm{cost}(y, S)$ and $\mathrm{cost}(y, S) \ge (1 - 3\varepsilon)\,\mathrm{cost}(x, S)$.

$$\begin{aligned}\mathrm{cost}(C_i \cap \Omega, S) &= \sum_{x \in C_i \cap \Omega} \frac{|C_i|\,\mathrm{cost}_\tau(G, A)}{|\Omega|\,\mathrm{cost}_\tau(C_i, A)} \mathrm{cost}(x, S) \\ &= (1 \pm 2\varepsilon) \sum_{x \in C_i \cap \Omega} \frac{|C_i|\,\mathrm{cost}_\tau(G, A)}{|\Omega|\,\mathrm{cost}_\tau(C_i, A)} \mathrm{cost}(y, S) \\ &= (1 \pm 2\varepsilon)(1 \pm \varepsilon)|C_i|\,\mathrm{cost}(y, S) \\ &= (1 \pm 2\varepsilon)(1 \pm \varepsilon) \sum_{x \in C_i} \mathrm{cost}(y, S) \\ &= (1 \pm 2\varepsilon)(1 \pm \varepsilon)(1 \pm 3\varepsilon)\,\mathrm{cost}(C_i, S) \\ &= (1 \pm O(\varepsilon))\,\mathrm{cost}(C_i, S).\end{aligned}$$

The third equation holds because of event $\mathcal{E}$. □

Proof of Lemma B.11. Let $x_{i,S} := \arg\min_{x \in C_i} \mathrm{cost}(x, S)$ and $w_{x,S} = (\mathrm{cost}(x, S) - \mathrm{cost}(x_{i,S}, S)) \,/\, \mathrm{cost}_\tau(x_{i,S}, A)$. Let $E_{l,S} := \sum_{C_i \in L_S} \sum_{x \in C_i \cap I_{l,S} \cap \Omega} w(x)\,\mathrm{cost}_\tau(x_{i,S}, A) w_{x,S}$ and $F_{l,S} := \sum_{C_i \in L_S} \sum_{x \in C_i \cap I_{l,S} \cap \Omega} w(x)\,\mathrm{cost}(x_{i,S}, S)$.

$w_{x,S}$ is bound. For fixed i, l and S consider arbitrary $x \in C_i \cap I_{l,S}$. By the definition of $x_{i,S}$ it is straightforward to see $\mathrm{cost}(x, S) \ge \mathrm{cost}(x_{i,S}, S)$, and thus $w_{x,S} \ge 0$. Besides, because the property of well-structured group there are

$$\mathrm{cost}(x_{i,S}, S) \le \mathrm{cost}(x, S) \le 2^{l+1}\,\mathrm{cost}_\tau(x, A) \le 2^{l+2}\,\mathrm{cost}_\tau(x_{i,S}, A)$$

and

$$\mathrm{cost}(x, x_{i,S}) \le 2^{z-1}(\mathrm{cost}(x, A) + \mathrm{cost}(x_{i,S}, A)) \le 3 \cdot 2^{z-1}\,\mathrm{cost}_\tau(x_{i,S}, A).$$

Using Lemma B.2, for any $\alpha \le 1$,

$$\mathrm{cost}(x, S) \le (1 + \frac{\alpha}{z})^{z-1}\,\mathrm{cost}(x_{i,S}, S) + (1 + \frac{z}{\alpha})^{z-1}\,\mathrm{cost}(x, x_{i,S})$$

which after rearranging implies

$$\begin{aligned}\mathrm{cost}(x, S) - \mathrm{cost}(x_{i,S}, S) &\le 2\alpha\,\mathrm{cost}(x_{i,S}, S) + (\frac{2z}{\alpha})^{z-1}\,\mathrm{cost}(x, x_{i,S}) \\ &\le 2^z(2\alpha \max(1, 2^{l+1}) + (\frac{2z}{\alpha})^{z-1})\,\mathrm{cost}_\tau(x_{i,S}, A)\end{aligned}$$

Let $\alpha = 2^{-l/z}$ (ignoring constants that depend on z), the inequality yields that

$$\mathrm{cost}(x, S) - \mathrm{cost}(x_{i,S}, S) \le 2^{O(z\log z)} 2^{l(1-1/z)}\,\mathrm{cost}_\tau(x_{i,S}, A).$$

Therefore, $w_{x,S} \in [0, 2^{O(z\log z)}2^{l(1-1/z)}]$.

$E_{l,S}$ can be expressed differently:

$$E_{l,S} = \sum_{C_i\in L_S}\sum_{x\in C_i\cap I_{l,S}\cap\Omega} w(x)\operatorname{cost}_\tau(x_{i,S},A)w_{x,S} \tag{4}$$

$$= \sum_{C_i\in L_S}\sum_{x\in C_i\cap I_{l,S}\cap\Omega} w(x)(\operatorname{cost}(x,S)-\operatorname{cost}(x_{i,S},S)) \tag{5}$$

$$= \sum_{x\in I_{l,S}\cap L_S\cap\Omega} w(x)\operatorname{cost}(x,S) - F_{l,S}. \tag{6}$$

The expectation of $E_{l,S}$ is as follows:

$$\begin{aligned}\mathbb{E}[E_{l,S}] &= \sum_{x\in I_{l,S}\cap L_S} |\Omega|\Pr[x]w(x)\operatorname{cost}(x,S) - \mathbb{E}[F_{l,S}]\\ &= \sum_{x\in I_{l,S}\cap L_S} \frac{|\Omega|\operatorname{cost}_\tau(C_i,A)}{|C_i|\operatorname{cost}_\tau(G,A)}\frac{|C_i|\operatorname{cost}_\tau(G,A)}{|\Omega|\operatorname{cost}_\tau(C_i,A)}\operatorname{cost}(x,S) - \mathbb{E}[F_{l,S}]\\ &= \operatorname{cost}(I_{l,S}\cap L_S,S) - \mathbb{E}[F_{l,S}].\end{aligned}$$

Intuitively, by Bernstein's inequality the random variable $E_{l,S}$ is concentrated around its expectation. Let Ω_i be the point sampled from the i-th round of importance sampling (Line 9 in Algorithm 2), and let $X_i = w(\Omega_i)\operatorname{cost}_\tau(x_{\tau(\Omega_i),S},A)w_{\Omega_i,S}$ when $\Omega_i \in I_{l,S}\cap\Omega$ and $X_i = 0$ otherwise. It holds that $E_{l,S} = \sum_{i=1}^t X_i$. X_i and its variance are bounded.

$$\begin{aligned}\operatorname{Var}[X_i] \le \mathbb{E}[X_i^2] &= \sum_{x\in I_{l,S}\cap L_S}\Pr[x]\left(w(x)\operatorname{cost}_\tau(x_{\tau(x),S},A)w_{x,S}\right)^2\\ &\le \sum_{x\in I_{l,S}\cap L_S}\frac{\operatorname{cost}_\tau(C_i,A)}{|C_i|\operatorname{cost}_\tau(G,A)}\left(\frac{|C_i|\operatorname{cost}_\tau(G,A)}{|\Omega|\operatorname{cost}_\tau(C_i,A)}\operatorname{cost}_\tau(x,A)2^{l(1-1/z)}2^{O(z\log z)}\right)^2\\ &\le \sum_{x\in I_{l,S}\cap L_S}2^{2l(1-1/z)}2^{O(z\log z)}\frac{|C_i|\operatorname{cost}_\tau(G,A)}{|\Omega|^2\operatorname{cost}_\tau(C_i,A)}\operatorname{cost}_\tau^2(x,A)\\ &\le \sum_{x\in I_{l,S}\cap L_S}2^{2l(1-1/z)}2^{O(z\log z)}\frac{\operatorname{cost}_\tau(G,A)}{|\Omega|^2}\operatorname{cost}_\tau(x,A)\end{aligned}$$

which implies

$$\operatorname{Var}[X_i] \le \begin{cases}\dfrac{2^{O(z\log z)}\operatorname{cost}_\tau(G,A)\operatorname{cost}_\tau(G,A)}{|\Omega|^2}, & z=1\\ \dfrac{2^{O(z\log z)}2^{l(1-2/z)}\operatorname{cost}_\tau(G,A)\operatorname{cost}(I_{l,S},S)}{|\Omega|^2}, & z\ge 2\end{cases}$$

Besides, X_i has upper bound.

$$\begin{aligned}X_i &\le \frac{|C_i|\operatorname{cost}_\tau(G,A)}{|\Omega|\operatorname{cost}_\tau(C_i,A)}\operatorname{cost}_\tau(x,A)2^{l(1-1/z)}2^{O(z\log z)}\\ &\le 2^{l(1-2/z)}2^{O(z\log z)}\frac{\operatorname{cost}_\tau(G,A)}{|\Omega|}.\end{aligned}$$

By Bernstein's inequality,

$$\Pr[|E_{l,S}-\mathbb{E}[E_{l,S}]| \le \frac{\varepsilon}{z\log z/\varepsilon}\cdot(\operatorname{cost}_\tau(G,A)+\operatorname{cost}(I_{l,S},S))] \le \exp\left(-\frac{\min(\varepsilon^2,\varepsilon^z)t}{2^{O(z\log z)}\log^2(1/\varepsilon)}\right).$$

Denote $F_S := \sum_{l \le z\log(4z/\varepsilon)} F_{l,S}$. Condition on event $\mathcal{E}$, the value of F_S and its expectation are as follows:

$$\begin{aligned}
\mathbb{E}[F_S] &= \sum_{l \le z\log(4z/\varepsilon)} \sum_{C_i \in L_S} \sum_{x \in C_i \cap I_{l,S}} |\Omega| \Pr[x] w(x) \operatorname{cost}(x_{i,S}, S) \\
&= \sum_{C_i \in L_S} |C_i| \operatorname{cost}(x_{i,S}, S); \\
F_S &= \sum_{C_i \in L_S} \sum_{x \in C_i \cap \Omega} w(x) \operatorname{cost}(x_{i,S}, S) \\
&= \sum_{C_i \in L_S} \operatorname{cost}(x_{i,S}, S) \sum_{x \in C_i \cap \Omega} \frac{|C_i| \operatorname{cost}_\tau(G, A)}{|\Omega| \operatorname{cost}_\tau(C_i, A)} \\
&= (1 \pm \varepsilon) \sum_{C_i \in L_S} |C_i| \operatorname{cost}(x_{i,S}, S).
\end{aligned}$$

Hence $F_S = (1 \pm \varepsilon)\mathbb{E}[F_S]$, and $\mathbb{E}[F_S] \le \operatorname{cost}(L_S, S) \le \operatorname{cost}(G, S)$.

Taking an union bound over the concentration for all possible $S \in \mathbb{C}^k$ and all l such that $\log(\varepsilon/2) \le l \le z\log(4z/\varepsilon)$, it holds with probability $1 - \exp\left(k\log(|\mathbb{C}|) - 2^O(z\log z) \cdot \min(\varepsilon^2, \varepsilon^z) \cdot t \cdot \log^{-2}(1/\varepsilon)\right)$ that, for every $S \in \mathbb{C}^k$ and $\log(\varepsilon/2) \le l \le z\log(4z/\varepsilon)$,

$$|E_{l,S} - \mathbb{E}[E_{l,S}] \le \frac{\varepsilon}{z\log(z/\varepsilon)} \left(\operatorname{cost}_\tau(G, A) + \operatorname{cost}(I_{l,S}, S)\right)|.$$

Conditioning on the above event together with event $\mathcal{E}$, It holds that

$$\begin{aligned}
|\operatorname{cost}(L_S, S) - \operatorname{cost}(L_S \cap \Omega, S)| &= |\sum_{x \in L_S} \operatorname{cost}(x, S) - \sum_{x \in L_S \cap \Omega} w(x) \operatorname{cost}(x, S)| \\
&\le |\sum_{x \in L_S} \operatorname{cost}(x, S) - \mathbb{E}[F_S] + F_S - \sum_{x \in L_S \cap \Omega} w(x) \operatorname{cost}(x, S)| + |\mathbb{E}[F_S] - F_S| \\
&\le \sum_{l < \log(\varepsilon/2)} |\sum_{x \in I_{l,S} \cap L_S} \operatorname{cost}(x, S) - \mathbb{E}[F_{l,S}] + F_{l,S} - \sum_{x \in I_{l,S} \cap L_S \cap \Omega} w(x) \operatorname{cost}(x, S)| \\
&\quad + \sum_{l = \log(\varepsilon/2)}^{z\log(4z/\varepsilon)} |\sum_{x \in I_{l,S} \cap L_S} \operatorname{cost}(x, S) - \mathbb{E}[F_{l,S}] + F_{l,S} - \sum_{x \in I_{l,S} \cap L_S \cap \Omega} w(x) \operatorname{cost}(x, S)| \\
&\quad + |\mathbb{E}[F_S] - F_S|.
\end{aligned}$$

The tiny ranges can be bound as follows since $F_{l,S} \le \sum_{x \in I_{l,S} \cap \Omega} w(x) \operatorname{cost}(x, S)$ and $\mathbb{E}[F_{l,S}] \le \sum_{x \in I_{l,S}} \operatorname{cost}(x, S)$:

$$\begin{aligned}
&\sum_{l < \log(\varepsilon/2)} |\sum_{x \in I_{l,S} \cap L_S} \operatorname{cost}(x, S) - \mathbb{E}[F_{l,S}] + F_{l,S} - \sum_{x \in I_{l,S} \cap L_S \cap \Omega} w(x) \operatorname{cost}(x, S)| \\
\le &\sum_{l < \log(\varepsilon/2)} \left(\sum_{x \in I_{l,S} \cap L_S} \operatorname{cost}(x, S) + \mathbb{E}[F_{l,S}] + F_{l,S} + \sum_{x \in I_{l,S} \cap L_S \cap \Omega} w(x) \operatorname{cost}(x, S) \right) \\
\le &2 \left(\sum_{x \in I_{l,S}} \operatorname{cost}(x, S) + \sum_{I_{l,S} \cap \Omega} w(x) \operatorname{cost}(x, S) \right) \\
\le &4\varepsilon \operatorname{cost}_\tau(G, A).
\end{aligned}$$

Plugging this result into the previous inequality, it holds that

$$\begin{aligned}
&|\operatorname{cost}(L_S, S) - \operatorname{cost}(L_S \cap \Omega, S)| \\
\leq& 4\varepsilon \operatorname{cost}_\tau(G, A) + \sum_{l=\log(\varepsilon/2)}^{z\log(4z/\varepsilon)} |E_{l,S} - \mathbb{E}[E_{l,S}]| + |\mathbb{E}[F_S] - F_S| \\
\leq& 4\varepsilon \operatorname{cost}_\tau(G, A) + (z\log(4z/\varepsilon) - \log(\varepsilon/2)) \cdot \frac{\varepsilon}{z\log(z/\varepsilon)} (\operatorname{cost}_\tau(G, A) + \operatorname{cost}(L_S, S)) + \varepsilon \operatorname{cost}(G, S) \\
\leq& O(\varepsilon)(\operatorname{cost}_\tau(G, A) + \operatorname{cost}(G, S)).
\end{aligned}$$

□

Outer groups For outer groups, the following lemma holds:

Lemma B.12. *Let*

$$t = \tilde{O}\left(2^{O(z)} \cdot m \cdot (d + \log n) \cdot \frac{1}{\varepsilon^2}\right)$$

in Algorithm 2 Line 10. It holds with probability at least $1/12$ *that, for any group of outer rings* $G = G_b^O$ *and the corresponding sample* Ω_b^O*, and for any* $S \in (\mathbb{R}^d)^m$*,*

$$|\operatorname{cost}(G, S) - \operatorname{cost}(\Omega, S)| \leq 2\frac{c_\tau^z \varepsilon}{z\log(z/\varepsilon)}(\operatorname{cost}(D, S) + \operatorname{cost}(D, A)).$$

Proof. Fix an arbitrary S. Partition the points in G into two parts and denote

$$\begin{aligned}
G_{\text{close},S} &:= \{x \in G \mid \operatorname{cost}(x, S) \leq 4^z \operatorname{cost}_\tau(x, A)\} \\
G_{\text{far},S} &:= \{x \in G \mid \operatorname{cost}(x, S) > 4^z \operatorname{cost}_\tau(x, A)\}.
\end{aligned}$$

Bernstein's inequality works for the close part. Let Ω_i be the i-th sampled point and let

$$X_i = \begin{cases} \dfrac{\operatorname{cost}_\tau(G, A)}{|\Omega| \operatorname{cost}_\tau(\Omega_i, A)} \cdot \operatorname{cost}(\Omega_i, S), & \Omega_i \in G_{\text{close},S} \\ 0, & \Omega_i \notin G_{\text{close},S} \end{cases}$$

Let $E_{\text{close},S} := \sum_{i=1}^t X_i$. The variance of X_i has the property

$$\begin{aligned}
\operatorname{Var}[X_i] \leq \mathbb{E}[X_i^2] &= \sum_{x \in G_{\text{close},S}} \left(\frac{\operatorname{cost}_\tau(G, A)}{|\Omega| \operatorname{cost}_\tau(x, A)} \operatorname{cost}(x, S)\right)^2 \frac{\operatorname{cost}_\tau(x, A)}{\operatorname{cost}_\tau(G, A)} \\
&= \frac{\operatorname{cost}_\tau(G, A)}{|\Omega|^2} \sum_{x \in G_{\text{close},S}} \frac{\operatorname{cost}(x, S)}{\operatorname{cost}_\tau(x, A)} \operatorname{cost}(x, S) \\
&\leq \frac{4^z}{|\Omega|^2} \operatorname{cost}_\tau(G, S) \operatorname{cost}(G, S).
\end{aligned}$$

X_i has a upper bound

$$X_i \leq \max_{x \in G_{\text{close},S}} \left(\frac{\operatorname{cost}_\tau(G, A)}{|\Omega| \operatorname{cost}_\tau(x, A)} \operatorname{cost}(x, S)\right) \leq \frac{4^z}{|\Omega|} \operatorname{cost}_\tau(G, A).$$

Bernstein's inequality yields that

$$\Pr[|E_{\text{close},S} - \mathbb{E}[E_{\text{close},S}]| \leq \frac{\varepsilon}{z\log(z/\varepsilon)}(\operatorname{cost}_\tau(D, A) + \operatorname{cost}(D, S))] \leq \exp\left(-2^{O(z)} \cdot \left(\frac{\varepsilon}{z\log(z/\varepsilon)}\right)^2 \cdot t\right)$$

Similar technique about A-approximate centroid set yields that for any S and any G,

$$|\operatorname{cost}(G_{\text{close},S}, S) - \operatorname{cost}(G_{\text{close},S} \cap \Omega, S)| \leq \frac{c_\tau^z \varepsilon}{z\log(z/\varepsilon)} (\operatorname{cost}(D, A) + \operatorname{cost}(D, S))$$

with probability at least

$$1-\exp\left(m\log n+O(md)\log(\frac{z^2}{\varepsilon}\log(z/\varepsilon))-2^{-O(z)}\varepsilon^2 t\right).$$

The proof for the far part is as follows. Denote event $\mathcal{E}_{\mathrm{far}}$ to be: for any cluster C,

$$\sum_{x\in C\cap G\cap\Omega} w(x)\operatorname{cost}_\tau(x,A)=(1\pm\varepsilon)\operatorname{cost}_\tau(C\cap G,A).$$

Event $\mathcal{E}_{far}$ happens with probability at least $1-m\exp(\varepsilon^2 t/m)$. Let $E_C=\sum_{i=1}^t X_i$, where

$$X_i=\begin{cases} w(x)\operatorname{cost}_\tau(\Omega_i,A), & \Omega_i\in C\cap G\\ 0, & \Omega_i\notin C\cap G\end{cases}$$

with Ω_i being the i-th sampled point. Calculation shows that $\mathrm{Var}[X_i]\le E[X_i^2]\le 2m\operatorname{cost}_\tau^2(C\cap G,A)/t^2$ and $X_i\le 2m\operatorname{cost}_\tau(C\cap G,A)/t$, and the Bernstein's inequality implies the success probability.

Fix a cluster C_i such that $C_i\cap G_{\mathrm{far},S}\neq\varnothing$ and let a_i be the center. Let x_i be a point such that $x_i\in C_i\cap G_{\mathrm{far},S}$, which implies $\operatorname{dist}(x_i,S)\ge 4\operatorname{dist}(x_i,a_i)$. Let $C_{\mathrm{close}}:=\{x\in C_i\mid \operatorname{cost}_\tau(x,A)\le(z/\varepsilon)^z(\operatorname{cost}_\tau(C_i,A)/|C_i|)\}$. Due to Markov's inequality, $|C_{\mathrm{close}}|\ge(1-\varepsilon/z)|C_i|$. Note that G is an outer group and for any $x\in G$ it holds that $\operatorname{cost}_\tau(x,A)\ge(z/\varepsilon)^{2z}\cdot(\operatorname{cost}_\tau(C_i,A)/|C_i|)$. By the definition of $G_{\mathrm{far},S}$ and Lemma B.2 it can be derived that

$$\begin{aligned}
\operatorname{cost}(a_i,S)&\ge(\operatorname{dist}(x_i,S)-\operatorname{dist}(x_i,a_i))^z\ge 3^z\operatorname{cost}_\tau(x_i,A)\ge 3^z(z/\varepsilon)^{2z}\frac{\operatorname{cost}_\tau(C_i,A)}{|C_i|}\\
\operatorname{cost}(a_i,S)&\le(1+\varepsilon)\operatorname{cost}(x,S)+(1+2z/\varepsilon)^{z-1}\operatorname{cost}(x,a_i)\\
&\le(1+\varepsilon)\operatorname{cost}(x,S)+(3z/\varepsilon)^{z-1}(z/\varepsilon)^z\frac{\operatorname{cost}_\tau(C_i,A)}{|C_i|}\\
&\le(1+\varepsilon)\operatorname{cost}(x,S)+\varepsilon\operatorname{cost}_\tau(a_i,S)
\end{aligned}$$

where x is an arbitrary point in C_{close}. Combining the lower bound of the size of C_{close}, this implies

$$\operatorname{cost}(C_i,S)\ge\operatorname{cost}(C_{\mathrm{close},S})|C_{\mathrm{close}}|\cdot\frac{1-\varepsilon}{1+\varepsilon}\cdot\operatorname{cost}(a_i,S)\ge 3^z(z/\varepsilon)^{2z-1}\operatorname{cost}_\tau(C_i,A).$$

Due to Markov's inequality the size of $G\cap C_i$ is bound by $(\varepsilon/z)^2|C_i|$ since G is an outer group. Combining with the above inequalities, it holds that

$$\begin{aligned}
\operatorname{cost}(G_{\mathrm{far},S}\cap C_i,S)&=\sum_{x\in G_{\mathrm{far},S}\cap C_i}\operatorname{cost}(x,S)\\
&\le\sum_{x\in G_{\mathrm{far},S}\cap C_i}(1+\varepsilon)\operatorname{cost}(a_i,S)+(1+2z/\varepsilon)^{z-1}\operatorname{cost}_\tau(G_{\mathrm{far},S}\cap C_i,A)\\
&\le\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^z\left(\frac{\varepsilon}{z}\right)^2\operatorname{cost}(C_i,S)+\left(\frac{3z}{\varepsilon}\right)^{z-1}\left(\frac{1}{3}\right)^z\left(\frac{\varepsilon}{z}\right)^{2z-1}\operatorname{cost}(G_{\mathrm{far},S}\cap C_i,S).
\end{aligned}$$

By simplifying the inequality and summing over all the clusters C_i, it is implied that

$$\operatorname{cost}(G_{\mathrm{far},S},S)\le\frac{\varepsilon}{z\log(z/\varepsilon)}\operatorname{cost}(D,S).$$

Condition on $\mathcal{E}_{\mathrm{far}}$, calculation shows that

$$\operatorname{cost}(G_{\mathrm{far},S}\cap\Omega,S)\le\frac{\varepsilon}{z\log(z/\varepsilon)}\operatorname{cost}(D,S).$$

The combination of the results about the close part and the far part tells that

$$\begin{aligned}
&|\operatorname{cost}(G,S)-\operatorname{cost}(G\cap\Omega,S)|\\
\le&|\operatorname{cost}(G_{\mathrm{close},S},S)-\operatorname{cost}(G_{\mathrm{close},S}\cap\Omega,S)|+|\operatorname{cost}(G_{\mathrm{far},S},S)|+|\operatorname{cost}(G_{\mathrm{far},S}\cap\Omega,S)|\\
\le&\frac{2c_\tau^z\varepsilon}{z\log(z/\varepsilon)}(\operatorname{cost}(D,S)+\operatorname{cost}(D,A)).
\end{aligned}$$

To make the above inequality holds with probability at least $1/12$, it is sufficient to set

$$t = \tilde{O}\left(2^{O(z)} \cdot m \cdot (d + \log n) \cdot \frac{1}{\varepsilon^2}\right).$$

□

C. Proof of Multidimensional Quantum Counting

This section provides a detailed proof of Theorem 4.4 and Theorem 4.2.

Theorem 4.4 is restated as follows:

Theorem C.1. *Given two integers $1 \le m \le n$, two parameters $\varepsilon \in (0, 1/3)$, and a partition $\tau\colon [n] \to [m]$. For each $j \in [m]$, denote $D_j := \{i \in [n]\colon \tau(i) = j\}$ as the j-th part and $n_j := |D_j|$ for the size. Assume that we have an oracle $O_\tau\colon |i\rangle|0\rangle \to |i\rangle|\tau(j)\rangle\ \forall i \in [n]$. For $\varepsilon \in (0, 1/3)$, $\delta > 0$, Algorithm 3 outputs $\tilde{n}_j$ such that $|\tilde{n}_j - n_j| \le \varepsilon n_j$ for every $j \in [m]$ with probability at least $1-\delta$, using $\tilde{O}\left(\sqrt{nm/\varepsilon}\log(1/\delta)\right)$ queries to O_τ, $\tilde{O}\left((\sqrt{nm/\varepsilon} + m/\varepsilon)\log(n/\delta)\log M\right)$ gate complexity, and additional $O(m \log M)$ classical processing time. The query complexity is optimal up to a logarithm factor.*

Proof. We establish the correctness and complexity bound separately.

Correctness The "maximal total probability" p_{mt} and precision $\frac{2\tilde{n}\varepsilon}{3nm}$ in Line 6 satisfies Lemma 4.5. Denote the exact cardinality of Q in Line 13 as n', $|\tilde{n} - n'| \le \frac{1}{2}n'$. For the maximal total probability, we have $\sum_{j\in S} p_j = \sum_{j\in S} \frac{n_j}{n} = \frac{n'}{n} \le \frac{2\tilde{n}}{n} = p_{mt}$. And for the precision, $\frac{2\tilde{n}\varepsilon}{3nm} \le \frac{n'}{nm}\varepsilon \le \varepsilon < 1/3$.

For each $j \in M$, n_j has been estimated. For those n_j estimated after the while loop stops, we find all the members belonging to the corresponding subset and count classically in Line 16, which gives an exact cardinality.

For those n_j estimated in the while loop, $|\tilde{p_j} - \frac{n_j}{n}| = |\tilde{p_j} - p_j| \le \frac{2\tilde{n}\varepsilon}{3nm}$. Since $\tilde{p_j} \ge \frac{\tilde{n}}{nm}$, $|\tilde{p_j} - \frac{n_j}{n}| \le \frac{2}{3}\varepsilon\tilde{p_j}$, thus $|\tilde{n}_j - n_j| \le \frac{2}{3}\varepsilon\tilde{n}_j$. Since $\varepsilon \in (0, 1/3)$, $|\tilde{n}_j - n_j| \le \varepsilon n_j$ as required.

Complexity In each iteration, the estimation process for $\{p_j\}_{j\in P}$ in Line 6 uses $O(\sqrt{p_{mt}}/\frac{2\tilde{n}\varepsilon}{3nm}) = O(\frac{m\sqrt{n}}{\varepsilon\sqrt{\tilde{n}}}) = O(\frac{\sqrt{nm}}{\varepsilon})$ applications of U_p and membership queries for P, and $\tilde{O}(m + p_{mt}/\frac{2\tilde{n}\varepsilon}{3nm}) = \tilde{O}(\frac{m}{\varepsilon})$ gate complexity, since $\tilde{n} < m/\varepsilon$ (Line 15) at this time. The estimation for the cardinality of Q needs $\tilde{O}(\sqrt{n/m}/\varepsilon)$ membership queries to P according to Lemma 2.6. The classical process in Line 7-11 and Line 16 needs at most $O(m)$ time. Since we can make all elements in P sorted in $O(m\log(m))$ time to keep an $O(\log(m))$ query complexity for the membership query to P, the total query complexity per iteration is $\tilde{O}(\sqrt{\frac{nm}{\varepsilon}})$, with additional $\tilde{O}(\frac{m}{\varepsilon})$ processing time.

The loop (Line 5-15) has at most $O(\log n)$ iterations. Denote $\tilde{n}_0 = n$ and $\tilde{n}$ obtained in Line 13 at the t-th iteration as $\tilde{n}_t$. At the $(t+1)$-th iteration, for any $j \in P$ there is $\tilde{p_j} \le \frac{\tilde{n}_t}{9nm}$, thus $p_j \le \tilde{p_j} + \frac{2\tilde{n}_t\varepsilon}{3nm} \le \frac{\tilde{n}_t}{3nm}$. We can calculate that $\tilde{n}_{t+1} \le \frac{3}{2}n'_{t+1} = \frac{3n}{2}\sum_{j\in P} p_j \le \frac{3nm}{2}\max_{j\in P} p_j \le \frac{\tilde{n}_t}{2}$. As a result, after $O(\log n)$ iterations there must be $\tilde{n} \le m/\varepsilon$.

Finding all the items remaining in Q (Line 16) requires $\tilde{O}(\frac{\sqrt{nm}}{\varepsilon})$ queries to O_τ and membership queries to P since $|P| \le m/\varepsilon$ here. The overall query complexity is $\tilde{O}(\frac{\sqrt{nm}}{\varepsilon})$.

There are at most $O(\log n)$ iterations and each iteration fails with probability at most $O(\frac{\delta}{\log n})$. Therefore, the success probability is at least $1 - \delta$. □

Theorem 4.2 is restated as follows:

Theorem C.2 (Multidimensional Quantum Approximate Summation)**.** *Given two integers $1 \le m \le n$, a real parameter $\varepsilon > 0$, a partition $\tau\colon [n] \to [m]$, and a function $f\colon [n] \to \mathbb{R}_{\ge 0}$. Assume that there exists access to an oracle $O_\tau\colon |i\rangle|0\rangle|0\rangle \to |i\rangle|\tau(i)\rangle|f(i)\rangle\ \forall i \in [n]$ and assume that f has an upper bound M. For $\varepsilon \in (0, 1/3)$, $\delta > 0$, there exists a quantum algorithm that finds ε-estimation for each $s_j := \sum_{\tau(i)=j} f(i)$, $j \in [m]$ with probability at least $1 - \delta$, using $\tilde{O}\left(\sqrt{nm/\varepsilon}\log(1/\delta)\log M\right)$ queries to O_τ and additional $\tilde{O}\left((\sqrt{nm/\varepsilon} + m/\varepsilon)\log(n/\delta)\log M\right)$ gate complexity.*

Proof. Write $f(i)$ as a binary number $f_0(i)f_1(i)\dots f_l(i)$, $l=\lceil \log M\rceil$. For each $t=0:l$, let O_t be

$$O_t\colon\ |i\rangle|0\rangle \to |i\rangle|\tau(i)I(f_t(i)=1)\rangle\,,$$

where $I(f_t(i)=1)$ is the indicator for whether $f_t(i)=1$. O_t can be constructed by constant queries to O_τ and its inverse:

$$|i\rangle|0\rangle|0\rangle|0\rangle \xmapsto{O_\tau} |i\rangle|\tau(i)\rangle|f(i)\rangle|0\rangle \mapsto |i\rangle|\tau(i)\rangle|f(i)\rangle|\tau(i)I(f_t(i))\rangle \xmapsto{O_\tau^{-1}} |i\rangle|0\rangle|0\rangle|\tau(i)I(f_t(i)=1)\rangle\,.$$

Applying Theorem C.1 with oracle O_t and $\delta'=\delta/l$, Algorithm 3 outputs $\tilde{s}_j^t$ for $j=1:m$, $\tilde{s}_j^t$ is an ε-estimation for $s_j^t=\sum_{\tau(i)=j} f_t(i)$ using $\tilde{O}\left(\sqrt{nm/\varepsilon}\log(l/\delta)\right)$ calls for O_t and additional $\tilde{O}\left((\sqrt{nm/\varepsilon}+m/\varepsilon)\log(nl/\delta)\right)$ gate complexity. Let $\tilde{s}_j=\sum_{t=0}^{l}2^t\tilde{s}_j^t$.

$$|\tilde{s}_j-s_j|\le\sum_{t=0}^{l}2^t|\tilde{s}_j^t-s_j^t|\le\varepsilon\sum_{t=0}^{l}2^t s_j^t\le\varepsilon s_j.$$

Therefore, $\tilde{s}_j$ is an ε-estimation of s_j, $\forall j\in[m]$. The total complexity is $\tilde{O}\left(\sqrt{nm/\varepsilon}\log(1/\delta)\log M\right)$ queries to O_τ, $\tilde{O}\left((\sqrt{nm/\varepsilon}+m/\varepsilon)\log(n/\delta)\log M\right)$ gate complexity, and additional $O(m\log M)$ classical processing time. □

D. Proofs of Quantum Lower Bounds

D.1. Auxiliary Lemmas

In the proofs of our quantum lower bounds, we use the following tools.

Theorem D.1 (The Perfect Composition Theorem, Høyer et al. 2007, Lee et al. 2011, Kimmel 2013, and Reichardt 2014)**.** *For the alphabet set Σ,Γ, functions $f:D_1\to K$ and $g:\mathcal{D}_2\to\Gamma$ with $\mathcal{D}_1\subseteq\Gamma^n,\mathcal{D}_2\subseteq\Sigma^m$, let $f\bullet g=f(g^n)$. The bounded-error quantum query complexity Q satisfies*

$$Q(f\bullet g)=\Theta(Q(f)\cdot Q(g)).$$

Corollary D.2 (A Direct Sum Theorem)**.** *For the alphabet set Σ, functions $g:\mathcal{D}\to\{0,1\}$ with $\mathcal{D}\subseteq\Sigma^m$, the bounded-error quantum query complexity Q satisfies*

$$Q(g^n)=\Theta(nQ(g)).$$

Proof. Plug $f=\mathrm{id}$ in Theorem D.1. Then, we only need to prove that $Q(f)=\Omega(n)$. This can be seen, e.g., by reducing to PARITY, which is defined and proved to have $Q(\mathrm{PARITY})=n/2$ in Beals et al. (2001). □

Definition D.3. We define the following problems:

- Decisional Quantum Counting: $f_{n',l,l'}\colon S\to\{0,1\}$ where $S\subset\{0,1\}^{n'}, l\neq l'$ is a partial Boolean function defined as

$$f_{n',l,l'}(x_0,x_1,\dots,x_{n'-1})=\begin{cases}0 & \text{if } |X|=l\\ 1 & \text{if } |X|=l'\\ \text{not defined} & \text{otherwise}\end{cases}\tag{7}$$

  where $|X|=\sum_{i=0}^{n'-1}x_i$. The notation $f_{n',l,l'}^{(k)}\colon S^k\to\{0,1\}^k$ represents the repeated direct product of f, i.e. $f_{n',l,l'}^{(k)}\left(x^{(1)},x^{(2)},\dots,x^{(k)}\right)=\left(f_{n',l,l'}\left(x^{(1)}\right),f_{n',l,l'}\left(x^{(2)}\right),\dots,f_{n',l,l'}\left(x^{(k)}\right)\right)$.

- Approximate Bits Finding: $\mathrm{Approx}_k\subseteq\{0,1\}^k\times\{0,1\}^k$ is a relation problem where for input bits $x_1,\dots,x_k$, output bits $c_1,\dots,c_k$ are correct iff the hamming distance between x and c is less than $\varepsilon_0\cdot k$ for some absolute constant $0<\varepsilon_0<1$ to be determined in the proof of Theorem D.9.

- The operator $\circ$ composites a relation and a function in the natural way, resulting in a relation problem on $S^k\times\{0,1\}^k$.

Theorem D.4 (Theorem 1.3 of Nayak & Wu 1999)**.** *Any bounded-error quantum algorithm that computes $f_{n',l,l'}$, given the input as an oracle, must make $\Omega\left(\sqrt{n'/\Delta}+\sqrt{n'(n'-m)/\Delta}\right)$, where $\Delta:=|l-l'|$ and $m\in\{l,l'\}$ s.t. $|m-n'/2|$ is maximized.*

D.2. Proof of Theorem 5.1

Now, we give the proof of Theorem 5.1, which is restated below:

Theorem D.5 (Quantum Lower Bound for Multidimensional Counting)**.** *Every quantum algorithm that solves the multidimensional counting problem (Definition 4.3) w.p. at least $\frac{2}{3}$ uses at least $\Omega\left(\sqrt{nk}\varepsilon^{-1/2}\right)$ queries to O_τ.*

Proof. Let $T = \lfloor 0.1\varepsilon^{-1} \rfloor$. Assume M is even. We reduce from the problem $f^{m/2}_{n/m,T,T+1}$. By Theorem D.4 and Corollary D.2, $Q\left(f^m_{n/m,T,T+1}\right) = \Omega\left(\sqrt{nm}\varepsilon^{-1/2}\right)$. The reduction applies by defining $\tau_i = 2a + x_i + 1$ for $i = am/2 + b$ where $1 \le b \le m/2$, calling the quantum multidimensional counter to get $n_1, n_2, \ldots, n_m$, and outputting $n_2 - T, n_4 - T, \ldots, n_m - T$. □

D.3. Proof of Theorem 5.2

Now, we prove Theorem 5.2, which is restated below:

Theorem D.6 (Quantum Lower Bounds for k-means and k-median)**.** *Assume that ε is sufficiently small. Consider the Euclidean k-means/median problem on data set $D = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$. Assume a quantum oracle $O_x |i, b\rangle := |i, b \oplus x_i\rangle$. Then, every quantum algorithm outputs the followings with probability $2/3$ must have quantum query complexity lower bounds for the following problems:*

- *An ε-coreset: $\Omega\left(\sqrt{nk}\varepsilon^{-1/2}\right)$ for k-means and k-median (Theorem D.7);*
- *An ε-estimation to the value of the objective function: $\Omega\left(\sqrt{nk} + \sqrt{n}\varepsilon^{-1/2}\right)$ for k-means and k-median (Theorem D.8);*
- *A center set C such that $\mathrm{cost}(C) \le (1+\varepsilon)\,\mathrm{cost}\,(C^*)$ where C^* is the optimal solution: $\Omega\left(\sqrt{nk}\varepsilon^{-1/6}\right)$ for k-means; $\Omega\left(\sqrt{nk}\varepsilon^{-1/3}\right)$ for k-median (Theorem D.9).*

We prove these different settings separately as follows.

D.3.1. Coreset Output

Theorem D.7. *Assume that ε is sufficiently small. Consider the Euclidean (k, z)-clustering problem on data set $D = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$. An oracle $O_x |i, b\rangle := |i, b \oplus x_i\rangle$ is accessible. Then, every quantum algorithm that outputs an ε-coreset w.p. at least $\frac{2}{3}$ uses at least $\Omega\left(\sqrt{nk}\varepsilon^{-1/2}\right)$ queries to O_x.*

Proof. We reduce from the multidimensional counting problem. The instance is 1-dimensional. Let $B = n^{100}$. The reduction is simply defining $x_i = \tau_i \cdot B$. After getting an ε-coreset from the (m, z)-clustering solver, we are able to query an ε-estimate of $\mathrm{cost}_z(D, C_i)$ where $C_i = \{B, 2B, \ldots, i \cdot B + 1, mB\}$, which equals to $|D_i|$, completing the proof. □

D.3.2. Objective Function Estimation

Theorem D.8. *Assume that ε is sufficiently small. Consider the Euclidean (k, z)-clustering problem on data set $D = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$. An oracle $O_x |i, b\rangle := |i, b \oplus x_i\rangle$ is accessible. Then, every quantum algorithm that outputs a real number $\tilde{A} \in (1 \pm \varepsilon) \min_{C^*} \mathrm{cost}(C^*)$ w.p. at least $\frac{2}{3}$ uses at least $\Omega\left(\sqrt{nk} + \sqrt{n}\varepsilon^{-1/2}\right)$ queries to O_x.*

Proof. Our proof has two parts: $\Omega(\sqrt{nk})$ and $\Omega(\sqrt{n}\varepsilon^{-1/2})$.

First, we prove that the complexity is $\Omega(\sqrt{nk})$. We can assume that $k = o(n)$. We reduce from the problem $f_{n,k,k+1}$, which has lower bound $\Omega(\sqrt{nk})$ by Theorem D.4. The instance is one-dimensional. We set $x'_i = 0$ if $x_i = 0$ and $x'_i = i$ if $x_i = 1$. Then, we estimate the (k, z)-clustering objective function. In the 0-case, the objective function value must be 0 (we have k centers for k points s.t. $x_i = 1$); in the 1-case, the objective function value is greater than 0. Thus, an ε-approximation is able to distinguish two cases, completing this part.

Second, we prove that the complexity is $\Omega(\sqrt{n}\varepsilon^{-1/2})$, even for $k=1$. Let $T=\lfloor 0.1\varepsilon^{-1}\rfloor$. We may assume $\varepsilon^{-1}=o(n)$. We reduce from the $f_{n,T,T+1}$. The instance is one-dimensional. The reduction is simply setting $x_i'=x_i$. Let $a=\sum_{i=1}^n x_i$. We need to distinguish two cases: $a=T$ and $a=T+1$. The objective function is $f(x)=a|x|^z+(n-a)|1-x|^z$. Assume $z>1$. By calculus, we can see $\min_x f(x)=\frac{a(n-a)}{\left((n-a)^{1/(z-1)}+a^{1/(z-1)}\right)^{z-1}}$. Thus, to prove that an ε-estimation can distinguish two cases, we must prove that,

$$\frac{T(n-T)}{\left((n-T)^{1/(z-1)}+T^{1/(z-1)}\right)^{z-1}}<(1-\varepsilon)\frac{(T+1)(n-T-1)}{\left((n-T-1)^{1/(z-1)}+(T+1)^{1/(z-1)}\right)^{z-1}}.$$

Because $T=o(n)$, $\frac{\text{RHS}}{\text{LHS}}\sim(1-\varepsilon)\frac{T+1}{T}>1$ for sufficiently small ε. As for $z=1$, $f(x)=a$ so the above argument is also valid. □

D.3.3. Center Set Output

Theorem D.9. *Assume that ε is sufficiently small. Consider the Euclidean (k,z)-clustering problem on data set $D=\{x_1,\ldots,x_n\}\subset\mathbb{R}^d$. An oracle $O_x|i,b\rangle:=|i,b\oplus x_i\rangle$ is accessible where the second register saves the binary representation of a real number and can have any polynomial number of qubits. Then, every quantum algorithm that outputs the optimal centers $C=\{c_1\ldots,c_k\}$ such that $\text{cost}(C)\le(1+\varepsilon)\min_{C^*}\text{cost}(C^*)$ w.p. at least $\frac{2}{3}$ uses at least $\Omega\left(\min\left(n,\sqrt{nk}\varepsilon^{-1/3z}\right)\right)$ queries to O_x.*

Proof. For convenience, we assume that n is a multiple of k and focus on the $z=2$ case (the proof for the $z\neq 2$ case is similar). We only need to prove when $\frac{1}{\varepsilon^{-1/3}}=o(k)$. Let $T=\left\lfloor(16\varepsilon_0\varepsilon)^{-1/3}\right\rfloor-1$ where ε_0 is to be defined. We will reduce an instance of the $2k$-means problem from the problem $\mathcal{P}=\text{Approx}_k\circ f^{(k)}_{n/k,T,T+1}$.

Intuitively, solving the problem $\mathcal{P}$ need to solve k independent cases of a quantum counting problem (distinguishing l 1s from l' 1s), but only need to be correct on a constant fraction of instances. We prove that the quantum algorithm must cost k times the queries of the $k=1$ case, which can be lower bounded by Theorem D.4.

Now we describe the reduction. The hard instance is consist of k unit balls far apart and each unit ball has T or $T+1$ unit vectors and $n-T$ or $n-T+1$ origins on it. Let $\mathcal{A}$ be an optimal algorithm for the $2k$-means problem. The input of the problem $\mathcal{P}$ is $x\in S^k$ where $S=\{x\in\{0,1\}^{n/k}:|X|=T\text{ or }T+1\}$. Let $x=x_1x_2\ldots x_n$. We map each x_i to a point in $\mathbb{R}^d$ for $d=100\log k/\varepsilon^2$. Let $B=n^{100}$ and $i=ak+b$ for $0\le a<k, 1\le b\le n/k$. Define $v_i=(i\cdot B,0,\ldots,0)$. If $x_i=0$, we map it to the point v_{2a}; if $x_i=1$, we draw $t_i\leftarrow\{2,3,\ldots,d\}$ uniformly and randomly. Then we set the point as $v_{2a+1}+e_{t_i}$ where $\{e_1,e_2,\ldots,e_d\}$ is the standard basis of $\mathbb{R}^d$. Note that by the union bound and the Birthday Paradox, $\{t_i\}$ are distinct in each group of size n/k w.h.p. We condition on this from now on. Since the reduction is classical, simple, and local, we can implement the oracle for the $2k$-means problem directly. Finally, we call $\mathcal{A}$ on the above instance and output $(\text{round}(t(\|c_2-v_1\|_2)),\ldots,\text{round}(t(\|c_{2i}-v_{2i-1}\|_2)),\ldots,\text{round}(t(\|c_{2n}-v_{2n-1}\|_2)))$, where:

- $t(x)=\frac{1}{\frac{1}{\sqrt{T}}-\frac{1}{\sqrt{T+1}}}\left[\frac{1}{\sqrt{T}}-\min(\max(x,\frac{1}{\sqrt{T+1}}),\frac{1}{\sqrt{T}})\right]$ is a normalizing mapping;
- $\text{round}(x)=\begin{cases}0 & x\le\frac{1}{2}\\ 1 & x>\frac{1}{2}\end{cases}$.

By easy adjustments (we can assume that there is a point in each of $2k$ balls centering at v_i with radius $B/5$ because otherwise the cost is larger than $B/10$, which is very large; then, we can move each center to its nearest point on the unit ball centered at v_i), we can assume without loss of generality that the solutions outputted by the $2k$-means solver have the following properties:

1. $c_{2a-1}=v_{2a-2}$ for $1\le a\le k$;
2. $\|c_{2a},v_{2a-1}\|_2\le 1$.

Then, the cost of the clustering can be seen as the sum of costs of k independent 1-mean problems and each of the problem has size T or $T+1$. It is well known that, in the 1-mean problem of size n, $\mathrm{cost}(c) = \mathrm{cost}(c^*) + n\|c - c^*\|^2$ where $c^* = \frac{x_1+x_2+\cdots+x_n}{n}$ is the optimal center. We decompose $c_{2a} = v_{2a-1} + g_a$. Let $T_a = \{e_{t_i} : x_i = 1, (a-1)k < i \le ak\}$. (Recall that $|T_a| = T$ or $T+1$.) Define $g_a^* = \frac{1}{|T_a|}\sum_{e\in T_a} e$. Then, the clustering is ε-optimal if and only if

$$\sum_{a=1}^{k} |T_a| \|g_a - g_a^*\|^2 < \varepsilon \cdot \sum_{a=1}^{k} \frac{1}{2|T_a|} \sum_{x\neq y\in T_a} \mathrm{dist}(x,y)^2 \tag{8}$$

$$\implies \sum_{a=1}^{k} |T_a| \left(\|g_a\| - \|g_a^*\|\right)^2 < \varepsilon \cdot \sum_{a=1}^{k} (|T_a| - 1) \tag{9}$$

$$\implies T \cdot \sum_{a=1}^{k} \left(\|g_a\| - \frac{1}{\sqrt{|T_a|}}\right)^2 < \varepsilon \cdot k \cdot (T-1) \tag{10}$$

$$\implies \sum_{a=1}^{k} \left(\|g_a\| - \frac{1}{\sqrt{|T_a|}}\right)^2 < \varepsilon \cdot k. \tag{11}$$

Note that $t\left(\frac{1}{\sqrt{|T_a|}}\right)$ gives $f_{n/k,T,T+1}$ in the i-th block of size n/k, justifying the inner function of the problem $\mathcal{P}$.

$$\implies \left(\frac{1}{\sqrt{T}} - \frac{1}{\sqrt{T+1}}\right)^2 \sum_{a=1}^{k} \left(t(\|g_a\|) - t\left(\frac{1}{\sqrt{|T_a|}}\right)\right)^2 < \varepsilon \cdot k \tag{12}$$

$$\implies \frac{1}{4T\cdot(T+1)^2} \sum_{a=1}^{k} \left(t(\|g_a\|) - t\left(\frac{1}{\sqrt{|T_a|}}\right)\right)^2 < \frac{\varepsilon_0}{16(T+1)^3} \cdot k \tag{13}$$

$$\implies \sum_{a=1}^{k} \left(t(\|g_a\|) - t\left(\frac{1}{\sqrt{|T_a|}}\right)\right)^2 < \frac{\varepsilon_0}{4} \cdot k. \tag{14}$$

Intuitively, the k-means solver needs to solve k independent cases of $f_{n/k,T,T+1}$ where the right answers are $t\left(|T_a|^{-1/2}\right)$. $t(\|g_a\|) \in [0,1]$ are a fractional guess in $[0,1]^k$ of $\{0,1\}^k$. For convenience, we round the output. A simple lemma is required to bound the error of rounding:

Lemma D.10. *Given $x_1, x_2, \ldots, x_k \in [0,1]$ and $y_1, y_2, \ldots, y_k \in \{0,1\}$, we have that*

$$\sum_{i=1}^{k} \mathbf{1}_{\mathrm{round}(x_i)\neq y_i} \le 4 \sum_{i=1}^{k} (x_i - y_i)^2$$

.

Proof. One has:

$$\begin{aligned}
&\sum_{i=1}^{k} (x_i - y_i)^2 \\
=&\sum_{\mathrm{round}(x_i)\neq y_i} (x_i - y_i)^2 + \sum_{\mathrm{round}(x_i)= y_i} (x_i - y_i)^2 \\
\ge&\frac{1}{4} \sum_{t(x_i)\neq y_i} 1.
\end{aligned}$$

□

Back to the proof of Theorem D.9. Plugging $x_i = t(\|g_a\|)$ and $y_i = t\left(\frac{1}{\sqrt{|T_a|}}\right)$ in the above lemma, we have that the hamming distance between the output of our solver for $\mathcal{P}$ and $f^{(k)}_{n/k,T,T+1}$ is less than $\varepsilon_0 k$, so it is indeed a solver for the relation problem $\mathcal{P}$.

Now it is sufficiently to prove the lower bound for the problem $\mathcal{P}$. Consider another problem defined simply as $\mathcal{P}' = f^{(k)}_{n/k,T,T+1}$. Applying Corollary D.2, we have that $Q(\mathcal{P}') = kQ(f_{n/k,T,T+1})$. By Theorem D.4, $Q(f_{n/k,T,T+1}) = \Theta\left(\sqrt{n/kT}\right)$. Hence, $Q(\mathcal{P}') = \Theta\left(\sqrt{nk}\varepsilon^{-1/6}\right)$.

Thus, there exists a constant $C_0 > 0$ such that every quantum algorithm that solves $\mathcal{P}'$ w.p. at least $\frac{2}{3}$ uses at least $C_0\sqrt{nk}\varepsilon^{-1/6}$ queries. Let $\mathcal{A}$ query the oracle for t times. We construct an algorithm $\mathcal{A}'(x)$ for $\mathcal{P}'$ from $\mathcal{A}$: $\mathcal{A}'$ calls $c \leftarrow \mathcal{A}(x)$ and then uses the Grover search (Lemma 2.7) to find the set $S = \{i \in [k] : f^{(k)}_{n/k,T,T+1}(x)_i \neq c_i\}$ and then flips the bits of c in S and output c. By the definition of $\mathcal{P}'$, $|S| \leq \varepsilon_0 \cdot k$. And $f^{(k)}_{n/k,T,T+1}(x)_i$ can be computed by $\frac{\pi}{2}\sqrt{nT/k} + n/k \leq 2\sqrt{nT/k}$ queries by the Grover search too. Thus, finding S costs $2(\frac{\pi}{2}\sqrt{\varepsilon_0 \cdot k \cdot k} + \varepsilon_0 \cdot k)\sqrt{nT/k} \leq 100\varepsilon_0^{5/6}\sqrt{nk}\varepsilon^{-1/6}$ queries. We then have $t + 100\varepsilon_0^{5/6}\sqrt{nk}\varepsilon^{-1/6} \geq C_0\sqrt{nk}\varepsilon^{-1/6}$. Now set $\varepsilon_0 = \left(\frac{C_0}{1000}\right)^{6/5}$ and solve the inequality, we get $t \geq 0.9C_0\sqrt{nk}\varepsilon^{-1/6}$, completing the proof. □