Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning

Nai-Hui Chia* András Gilyén† Tongyang Li‡ Han-Hsuan Lin*
Ewin Tang§ Chunhao Wang*

Abstract

We present an algorithmic framework generalizing quantum-inspired polylogarithmic-time algorithms on low-rank matrices. Our work follows the line of research started by Tang's breakthrough classical algorithm for recommendation systems [STOC'19]. The main result of this work is an algorithm for singular value transformation on low-rank inputs in the quantum-inspired regime, where singular value transformation is a framework proposed by Gilyén et al. [STOC'19] to study various quantum speedups. Since singular value transformation encompasses a vast range of matrix arithmetic, this result, combined with simple sampling lemmas from previous work, suffices to generalize all results dequantizing quantum machine learning algorithms to the authors' knowledge. Via simple black-box applications of our singular value transformation framework, we recover the dequantization results on recommendation systems, principal component analysis, supervised clustering, low-rank matrix inversion, low-rank semidefinite programming, and support vector machines. We also give additional dequantizations results on low-rank Hamiltonian simulation and discriminant analysis.

*Department of Computer Science, University of Texas at Austin. Email: {nai,linhh,chunhao}@cs.utexas.edu
†Institute for Quantum Information and Matter, California Institute of Technology. Email: agilyen@caltech.edu
‡Department of Computer Science, Institute for Advanced Computer Studies, and Joint Center for Quantum Information and Computer Science, University of Maryland. Research supported by IBM PhD Fellowship, QISE-NET Triplet Award (NSF DMR-1747426), and the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Quantum Algorithms Teams program. Email: tongyang@cs.umd.edu
§University of Washington. Email: ewint@cs.washington.edu

Contents

1 Introduction

Motivation. The field of quantum machine learning (QML) has generated many proposals for how quantum computers could exponentially speed up machine learning tasks, particularly after Harrow, Hassidim, and Lloyd's sparse matrix inversion algorithm running in time poly-logarithmic in input size [HHL09]. If any one of these proposals truly manifest in a practical exponential speedup, it could be the killer application motivating the development of scalable quantum computers [Pre18]. However, despite the wealth of papers suggesting quantum algorithms for various machine learning problems (e.g. principal component analysis [LMR14], cluster assignment and finding [LMR13], support vector machines [RML14], recommendation systems [KP17]), their quantum speedups are not as "strong" as, say, Shor's algorithm for factoring [Sho97], because it is unclear how to load the input into a quantum computer efficiently or conclude useful information from the quantum outputs [Aar15].

In 2018, Tang gave a classical analog to the quantum recommendation systems algorithm [Tan19], previously believed to be one of the strongest candidates for QML speedup for practical problems. One of the biggest implications of Tang's breakthrough result is that its techniques appear to generalize to "dequantize" a wide swathe of QML algorithms, including principal component analysis and supervised clustering [Tan18], linear system solving [GLT18, CLW18], semidefinite program solving [CLLW19], support vector machines (SVM) [DBH19], nonnegative matrix factorization [CLS+19], minimal conical hull [DHLT19], etc. The framework that Tang used is a sampling-based model of the input matrices and vectors that replicates known QML algorithms while running on a classical computer in the regime where the inputs are low-rank matrices. In short, "dequantized" algorithms either provide strong barriers for or completely disprove the existence of exponential speedups from their corresponding QML algorithms in low-rank settings, which is a practical assumption in many of the applications.

Main results. A central goal of the research into "quantum-inspired" classical machine learning is to guide quantum machine learning research in the future. However, the previous research in this topic focuses on particular problems and only describes the particular tools that are necessary in each case. This work gives a description of the framework of quantum-inspired classical algorithms in *large generality*, exploring the capabilities and limitations of these techniques.

Specifically, our framework assumes a sample and query model where for a vector $v \in \mathbb{C}^n$, we can query v_i for a given $i \in [n]$ or sample a $j \in [n]$ with probability proportional to $|v_j|^2/\|v\|_2^2$ (see Definition 2.6); for a matrix $A \in \mathbb{C}^{m\times n}$, we denote by $\mathrm{SQ}(A)$ such a model for all its m rows. Given matrices $A^{(1)}, \dots, A^{(\tau)} \in \mathbb{C}^{m\times n}$ with $\mathrm{SQ}(A^{(1)}), \dots, \mathrm{SQ}(A^{(\tau)})$ and a Lipschitz continuous function[i] f, we achieve the following matrix arithmetic objective:

Main theorem (informal; see Theorem 3.1 and Theorem 3.3). *Let $A = UDV^\dagger$ be the singular value decomposition of $A = A^{(1)} + \cdots + A^{(\tau)}$ and let f be L-Lipschitz continuous. We can implement $\mathrm{SQ}(Uf(D)V^\dagger)$ up to ℓ_2-norm error ϵ in time $\widetilde{\mathcal{O}}\Big(\Big(\frac{\tau L^2(\sum_\ell \|A^{(\ell)}\|_F^2)}{\epsilon^2}\Big)^{18}\Big)$.*

Our core primitive is *singular value transformation* [GSLW19]. Roughly speaking, given a Hermitian matrix A with sample and query access $\mathrm{SQ}(A)$, along with a Lipschitz function f, we can achieve $\mathrm{SQ}(f(A))$ where f is applied to the singular values up to additive Frobenius norm error. Moreover, we can gain sample and query access to the decomposition of $f(A)$ into rank-1 matrices. This primitive has previously been noted to generalize a large portion of quantum machine learning research [GSLW19]; we bring this observation into the quantum-inspired landscape.

With our main theorem, we can recover existing quantum-inspired machine learning algorithms:

- Recommendation systems (Section 4.2): Given a matrix $A \in \mathbb{R}^{m\times n}$ with $\mathrm{SQ}(A)$, a row index $i \in [m]$, and a singular value threshold σ, the goal is to sample from the i^{th} row of a low-rank approximation of A which singular values $\geq \sigma$ with additive error $\epsilon\|A\|_F$. We apply the main theorem to a constant-Lipschitz continuous function f such that $f(x) = x$ on singular values in $[\frac{7}{6}\sigma, 1]$ and $f(x) = 0$ in $[0, \frac{5}{6}\sigma]$, which gives us the sample and query access to an approximated singular-value transformation of $f(A)$. Finally, we obtain a sample from the i^{th} row by the sampling techniques we have developed in Section 3.2. The running time is $\widetilde{\mathcal{O}}\Big(\frac{\|A\|_F^{24}}{\epsilon^{12}\sigma^{24}}\Big)$ (Corollary 4.5).

[i] The function here only needs to be Lipschitz continuous over an interval that contains all singular values of $A^{(1)} + \cdots + A^{(\tau)}$.

- Principal component analysis (Section 4.3): Given a matrix $X \in \mathbb{R}^{m\times n}$ with $\mathrm{SQ}(X)$ such that $\mathrm{rank}(X) = r$ and $X^T X$ has nonzero eigenvalues $\{\lambda_i\}_{i=1}^r$ and eigenvectors $\{v_i\}_{i=1}^r$ (without loss of generality $\lambda_1 \geq \cdots \geq \lambda_r$), the goal is to output λ_i up to additive error $\epsilon \operatorname{Tr}(X^T X)$ and $|v_i\rangle$ with probability $\lambda_i / \operatorname{Tr}(X^T X)$. This is in general impossible because distinguishing between λ_i and λ_{i+1} such that $\lambda_i - \lambda_{i+1} = O(1/\operatorname{poly}(n))$ necessarily takes $\operatorname{poly}(n)$ samples. However, if we know $K := \operatorname{Tr}(X^T X)/\lambda_k \geq k$ and $\eta := \min_{i\in[k]} |\lambda_i - \lambda_{i+1}|/\operatorname{Tr}(X^T X)$, then we can apply our main theorem to the function $f(x) = x^2$ to get an approximated singular-value decomposition of $X^T X$ and apply sampling access as a coupon collector problem; by doing that, we get all $\{\lambda_i\}_{i=1}^r$ and $\{\mathrm{SQ}(v_i)\}_{i=1}^r$ in time $\widetilde{\mathcal{O}}\Big(\frac{K}{(\epsilon\eta)^{18}}\Big)$ (Corollary 4.9).

- Supervised clustering (Section 4.4): Given a dataset of points $q_1, \ldots, q_m \in \mathbb{R}^n$ in $\mathbb{R}^n$, the goal is to estimate the distance between their centroid and a new point $p \in \mathbb{R}^n$, i.e., $\|p - \frac{1}{m}(q_1 + \cdots + q_m)\|^2$. We show how to use the sample and query access to estimate inner products: given $\mathrm{SQ}(M^T, w)$ where

$$M := \left[\frac{p}{\|p\|}, \frac{-q_1}{\|q_1\|}, \ldots, \frac{-q_m}{\|q_m\|}\right] \quad \text{and} \quad w := \left[\|p\|, \frac{\|q_1\|}{m}, \ldots, \frac{\|q_m\|}{m}\right]^T, \tag{1}$$

we approximate $\|p - \frac{1}{m}(q_1 + \cdots + q_m)\|^2$ to additive ϵ error in time $\mathcal{O}\big(\|M\|_F^2 \|w\| \frac{1}{\epsilon^2}\big)$ (Corollary 4.11).

- Matrix inversion (Section 4.5): Given a matrix $A \in \mathbb{R}^{n\times n}$ with $\mathrm{SQ}(A)$ and condition number κ, the goal is to obtain $\mathrm{SQ}(A^+)$ where A^+ is the pseudo-inverse of A. We apply our main theorem to an $\mathcal{O}(\kappa)$-Lipschitz function that is $1/x$ for $x \in [1/\kappa, 1]$ and 0 when $x \in [0, (1-\xi)/\kappa]$ for a $0 < \xi < 1$, and we get $\mathrm{SQ}(A^+)$ with ϵ-error in spectral norm in time $\widetilde{\mathcal{O}}\bigg(\Big(\frac{\|A\|_F^2 \kappa^4}{\epsilon^2 \xi^2}\Big)^{18}\bigg)$ (Theorem 4.12).

- Low-rank semidefinite programs (SDP) (Section 4.7): Given $m \in \mathbb{N}$, $b_1, \ldots, b_m \in \mathbb{R}$, and $n \times n$ Hermitian matrices $A^{(1)}, \ldots, A^{(m)}, C$ where $-I \preceq A^{(i)}, C \preceq I$ $\forall i \in [m]$ with their SQ access, denote OPT to be

$$\max \quad \operatorname{Tr}[CX] \tag{2}$$

$$\text{s.t.} \quad \operatorname{Tr}[A^{(i)}X] \leq b_i \quad \forall\, i \in [m];\ X \succeq 0. \tag{3}$$

The goal to output an X^* such that satisfying the above constraints while $\operatorname{Tr}[CX^*] \geq \mathsf{OPT} - \varepsilon$. It is known that SDP can be solved by the matrix multiplicative weight (MMW) method [AK07], where in iteration t we find a violated constraint j_t and take $X \leftarrow \exp[-\frac{\epsilon}{2}\sum_{\tau=1}^{t} A^{(j_\tau)}]$ (up to normalization). We apply our main theorem to the exponential function exp, and we obtain an ϵ-approximate optimal solution of the SDP as well as its SQ access in time $\widetilde{\mathcal{O}}(\operatorname{poly}(m, \tau, 1/\epsilon, 1/\sigma, 1/\eta))$ (Corollary 4.16).

- Support vector machines (Section 4.8): Given input data points $x_1, \ldots, x_m \in \mathbb{R}^n$ and their corresponding labels $y_1 \ldots, y_m = \pm 1$, let $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ be the specification of hyperplanes separating these points. The goal is to minimize the squared norm of the residuals:

$$\min_{w,b} \quad \frac{\|w\|^2}{2} + \frac{\gamma}{2}\|e\|^2 \tag{4}$$

$$\text{s.t.} \quad y_i(w^T x_i + b) = 1 - e(i), \quad \forall i \in [m], \tag{5}$$

where $e \in \mathbb{R}^m$ is a slack vector such that $e(j) \geq 0$ for $j \in [m]$. The dual of this problem is to maximize over the Karush-Kuhn-Tucker multipliers of a Lagrange function, taking partial derivatives of which yields a linear system:

$$\begin{bmatrix} 0 & \vec{1}^T \\ \vec{1} & X^T X + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \tag{6}$$

Therefore, solving this SVM can be regarded as solving a matrix inversion problem. Assuming $\mathrm{SQ}(X)$ and the minimum nonzero singular value of $X^T X$ is at least $m\epsilon_\kappa$, Theorem 4.12 implies that SVM can be solved with error ϵ in time $\widetilde{\mathcal{O}}\Big(\frac{\|X\|_F^{88}}{\epsilon^{26}\epsilon_\kappa^{72}}\Big)$ (Corollary 4.18).

These results we give can be applied in a black-box manner to recover the previous algorithmic results in this line of work, and thus are also conceptually simpler.

We also propose new quantum-inspired algorithm for other applications, including:

- Hamiltonian simulation (Section 4.1): Given a Hermitian matrix $H \in \mathbb{R}^{n\times n}$ with $\mathrm{SQ}(H)$ such that $\|H\| \leq 1$, a unit vector $b \in \mathbb{R}^n$ with $\mathrm{SQ}(b)$, and a time $t > 0$, the goal is to obtain $\mathrm{SQ}(v)$ where $\|v - e^{itH}b\|_F \leq \epsilon$. We apply our main theorem to the function $f(x) = e^{itx}$ (which is 2π-Lipschitz) and obtain $\mathrm{SQ}(e^{itH})$; we furthermore apply the matrix-vector product $e^{itH}b$ by a generalization of our main theorem (see Proposition 3.7). The final time complexity is $\widetilde{\mathcal{O}}\Big(\frac{t^{36}\|H\|_F^{36}}{\epsilon^{36}}\Big)$ (Corollary 4.2).

- Discriminant analysis (Section 4.6): Given M input data points $\{x_i \in \mathbb{R}^N : 1 \leq i \leq M\}$, each belonging to one of k classes. Let μ_c denote the centroid (mean) of class $c \in [k]$, and $\bar{x}$ denote the centroid of all data points. Let

$$S_B = \sum_{c=1}^{k} (\mu_c - \bar{x})(\mu_c - \bar{x})^T, \qquad S_W = \sum_{c=1}^{k} \sum_{x \in c} (\mu_c - x)(\mu_c - x)^T \tag{7}$$

  be the between-class scatter matrix and the weight matrix of the dataset, respectively. The goal is to find the largest p eigenvalues and eigenvectors of $S_W^{-1} S_B$. This can be regarded as a corollary of Theorem 4.12 and Theorem 3.9. Given $\mathrm{SQ}(X_{[k]})$, we apply our main theorem to an ϵ-approximation of the function $\frac{1}{\sqrt{x}}$ that is $\mathcal{O}(1/\epsilon)$-Lipschitz, with threshold θ (as in Theorem 4.12); the overall complexity is $\mathcal{O}\big(\mathrm{poly}(\|X\|_F, \epsilon^{-1}, \theta^{-1})\big)$.

Techniques. The fundamental algorithmic primitive we use is the sampling access SQ. This form of sampling is usually called *importance sampling* or *length-square sampling* in classical literature. In particular, we use the ability of importance sampling to approximate inner products of vectors (and higher-order tensors, more generally) and compute low-rank approximations (which, roughly speaking, follows from approximating matrix products).

We hope to revitalize study in importance sampling because of the developments of "quantum-inspired" machine learning algorithms lead to better understanding of existing classical problems and the space of problems that quantum algorithms can solve faster than their classical counterparts. Notably, *other types of sketches fail in the quantum-inspired model*: importance sampling takes time independent of dimension in the input regimes where quantum machine learning succeeds, whereas other randomized linear algebra methods such as Count-Sketch, Johnson-Lindenstrauss, and leverage score sampling all take time linear in input-sparsity. Furthermore, importance sampling is highly amenable to a quantum-like style of algorithms: given the ability to query entries and importance samples of the input, we can query entries and importance samples of the output, in the same way quantum machine learning algorithms move from an input quantum state to an output quantum state.

For example, this mode of thinking reveals that the low-rank approximation algorithm in [FKV04], which as stated requires $O(kmn)$ time to output the desired matrix, actually can produce useful results (samples and entries) in time independent of the input. Importance sampling is a weaker sketch compared to more modern sampling techniques, so our goal is to generalize previous results into a framework that demonstrates what can be done with it and establishes a possibility frontier for quantum algorithms to possibly push past.

Specifically, in [CLLW19] the method in [FKV04] has been extended in the context of matrix functions using the *symmetric approximation* technique. In this work, we further extend the technique to *asymmetric approximation* to accommodate the context of singular value transformations. To achieve this, we invoke the FKV algorithm twice, first to approximate the low-rank projection onto the left singular vector space and then to approximate the low-rank projection onto the right singular vector space. In this way, we have obtained a low-rank approximation of the original matrix in the form of $UU^\dagger DVV^\dagger$, where $U, V \in \mathbb{C}^{n\times r}$ are approximately isometries and $D \in \mathbb{C}^{r\times r}$ is a diagonal matrix. Here r is at most the rank of the original matrix and can be considered small. We do not have all the entries of U and V, but we have their sampling and query access. Note that $U^\dagger DV$ is a small matrix ($\in \mathbb{C}^{r\times r}$) whose entries can be estimated with a small error using sampling and query access of U and V. To perform the singular value transformation on $UU^\dagger DVV^\dagger$, we can apply the desired transformation f on $U^\dagger DV$, which is easy as it is a small matrix. Then the sampling and query access to $f(UU^\dagger DVV^\dagger)$ can be obtained using $f(U^\dagger DV)$ together with the sampling and query access to U and V.

Another technical difference from the previous work such as [CLW18, GLT18, CLLW19] is that our method *does not have an explicit dependence on the rank*. Instead, we use a slightly relaxed version of low-rank approximation following [Tan19, GSLW19]. This approximation transfers the dependence on the rank to the dependence on the Frobenius norm in the time complexity, and it allows our methods to be used for more general matrices (i.e., those have low-rank approximations, rather than are of strictly low-rank).

Prior work. Our work bridges the fields of randomized algorithms and quantum linear algebra. Generally speaking, the techniques our framework belong to randomized linear algebra methods (see the surveys [Mah11, KV17]).

We want to point out two other randomized linear algebra methods that also give sampling-based poly-logarithmic time classical algorithms for matrix arithmetics. The first is a generalization of [FKV04] from singular value decomposition to orthogonal tensor decomposition [DM07, MMD08], which can potentially be combined with our results on matrices to give their high-order counterparts. The second is the so-called Nyström method [WS01] that approximates a matrix by random projections onto its rows. A recent paper [RWC$^+$18] has shown how to use the Nyström method to simulate a sparse Hamiltonian using the sample and query access (Definition 2.6) in poly-logarithmic time, and it is a natural question to ask whether it can also be applied to other results in our paper.

It is also worthwhile to mention that although the degrees of the poly-logarithmic terms in quantum-inspired algorithms are not small, Ref. [ADBL19] conducted various numerical experiments and suggested that their performance in practice might work better than their theoretical guarantee.

In quantum linear algebra, similar papers have unified quantum machine learning results via simple primitives. Our primitive of singular value transformation is based in part on the quantum singular value transformation results of [GSLW19], but our techniques are significantly different when delve into dequantized, sampling-based classical algorithms. Some quantum linear algebra still remains untouched with these dequantization techniques, mostly because they are BQP-complete or seem to fundamentally use a reduction to a BQP-complete problem. For such algorithms, the problem typically becomes how to ensure that input matrix condition number stays small, the input can be quickly loaded into a quantum computer, and the output states can be used to conclude something meaningful (these are the caveats given by Aaronson in [Aar15]). Gaussian process regression [ZFF19] and topological data analysis [LGZ16] are examples of applications that tried to address these issues to get a super-polynomial quantum speedup.

Related independent work. Independently from our work, Jethwani, Le Gall and Singh simultaneously derived similar results. Similarly to our main results they also showed how to perform singular value transformation based on length-square sampling techniques, and how to apply the transformed matrices to an input vector – providing sample and query access to the output vector. Additionally to their results we also provide general matrix arithmetic primitives for adding and multiplying matrices having sample and query access. We also focus more on (re)deriving various dequantized algorithms.

Open questions. This work leaves several natural open questions for future investigation. In particular:

- Are there other quantum machine learning applications that can be dequantized by our framework? A natural proposal is topological data analysis [LGZ16].
- In the quantum setting, linear algebra algorithms [GSLW19] can achieve logarithmic dependence on error ϵ: why is this the case? Can classical algorithms also achieve such exponentially improved dependence, when the goal is restricted not to sample and query access, but just to sample access? If not, is there a mildly stronger classical model that can achieve this? Could this exponential speedup be instantiated in a meaningful way?
- Is there an approach to QML that does not go through [HHL09] (whose demanding assumptions make exponential speedups difficult to demonstrate even in theory) or a low-rank assumption (which we demonstrate makes machine learning easy for classical computers)?

Organization. The rest of the paper is organized as follows. We give preliminaries in Section 2. Our main technical results and their applications to dequantize quantum machine learning are presented in Section 3 and Section 4, respectively. We introduce singular value and spectral decompositions in Section 5 and their combinations with matrix functions in Section 6, and the proofs of the subsampling techniques are deferred to Appendix A. We also derive improved versions of the results in an alternative way in Appendix B, that achieve better complexities for even functions (Corollary B.4), and improve the polynomial degrees in the time complexities (Theorem B.7). The results in Appendix B can be used to optimize the complexities of our applications.

2 Preliminaries

To begin with, we define notations that will be used throughout this paper. We use $f \lesssim g$ to denote the ordering $f = \mathcal{O}(g)$, use $\widetilde{\mathcal{O}}(g(n))$ as a shorthand for $\mathcal{O}(g(n)\operatorname{poly}(\log n))$, and use $f \eqsim g$ to denote $f = g + o(g)$. For $z \in \mathbb{C}$, its absolute value is $|z| = \sqrt{z^* z}$, where z^* is the complex conjugate of z.

2.1 Linear algebra

In this paper, we consider complex matrices. Let $m, n \in \mathbb{N}$, and $A \in \mathbb{C}^{m\times n}$. We let $A(i,\cdot)$ be the i-th row, $A(\cdot,j)$ be the j-th column, and $A(i,j)$ is the (i,j)-th element of A. For vectors $v \in \mathbb{C}^n$, $\|v\|$ denotes standard Euclidean norm (so $\|v\| := (\sum_{i=1}^n |v_i|^2)^{1/2}$). For a matrix $A \in \mathbb{C}^{m\times n}$, the *Frobenius norm* of A is $\|A\|_F := (\sum_{i=1}^m \sum_{j=1}^n |A(i,j)|^2)^{1/2}$ and the *spectral norm* of A is $\|A\| := \sup_{x\in\mathbb{C}^n, \|x\|=1} \|Ax\|$, where $\|x\|$ is the Euclidean norm. We use $\operatorname{vec}(A) \in \mathbb{C}^{mn}$ to denote the vector formed by concatenating the rows of A.

We say A is an *isometry* if $\|Ax\| = \|x\|$ for all $x \in \mathbb{C}^n$. Equivalently, A is an isometry if $\langle A(\cdot,i), A(\cdot,j)\rangle = \delta_{ij}$ for $i,j \in [n]$. Here $\delta_{ij} = 1$ if $i = j$, and $\delta_{ij} = 0$ otherwise. It will also be useful to consider the notion of an *approximate isometry*.

Definition 2.1. Let $m, n \in \mathbb{N}$ and $m \geq n$. A matrix $V \in \mathbb{C}^{m\times n}$ is an *α-approximate isometry* if $\left\|V^\dagger V - I\right\| \leq \alpha$.

It is easy to check that an ϵ-approximate isometry has the following properties.

Remark 2.2. Let V be an ϵ-approximate isometry. The following statements hold:

1. $|\|V\| - 1| \leq \alpha$.
2. Let $A = UDV^\dagger$, and Π_A be the projector on the row space of A, then $\left\|VV^\dagger - \Pi_A\right\| \leq \alpha$.
3. There exists an isometry $U \in \mathbb{C}^{m\times n}$ whose column vectors span the column space o V satisfying $\|U - V\| \leq \alpha$.

For a matrix $A \in \mathbb{C}^{m\times n}$, let $N := \min(m,n)$. The *singular value decomposition* (SVD) of A is the representation $A = UDV^\dagger = \sum_{i=1}^N D(i,i)U(\cdot,i)V^\dagger(i,\cdot)$. Here, $U \in \mathbb{C}^{m\times N}$ and $V \in \mathbb{C}^{n\times N}$ are isometries, and $D \in \mathbb{R}^{N\times N}$ is diagonal with $\sigma_i := D(i,i)$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N \geq 0$.

We define the function ℓ_A to index the last singular value in $\{\sigma_1, \ldots, \sigma_r\}$ that is no less than the given value:

$$\ell_A(\sigma) := \max_i \{i \mid \sigma_i \geq \sigma\}. \tag{8}$$

We formally define singular value transformation:

Definition 2.3. Suppose that $A = \sum_{i=1}^r \sigma_i u_i v_i^\dagger$ is a singular value decomposition, and $f\colon [0,\infty) \to \mathbb{C}$ satisfies $f(0) = 0$, then we define the singular value transform of A as $f^{(\mathrm{SV})}(A) := \sum_{i=1}^r f(\sigma_i) u_i v_i^\dagger$.

Now, we define the low-rank approximation to A. The standard notion of low-rank approximation is that of $A_r := \sum_{i=1}^r \sigma_i U(\cdot,i)V(\cdot,i)^\dagger$, which is rank-$r$ matrix closest to A in spectral and Frobenius norm. We define analogous notions for focusing on singular values instead of rank using Eq. (8):

$$A_\sigma := \sum_{i=1}^{\ell_A(\sigma)} \sigma_i U(\cdot,i)V(\cdot,i)^\dagger := A\Pi_\sigma, \tag{9}$$

where Π_σ is defined as the projector onto the space spanned by $V(\cdot, 1), \ldots, V(\cdot, \ell_A(\sigma))$. We will need to relax this notion for our purposes, and introduce error $\eta \in [0, 1]$. In the following, we define $A_{\sigma,\eta}$ as in Section 2.1 of [Tan19].

Definition 2.4 ($A_{\sigma,\eta}$)**.** We define $A_{\sigma,\eta}$ as a singular value transform of A satisfying:

$$A_{\sigma,\eta} := P_{\sigma,\eta}^{(\mathrm{SV})}(A) \qquad P_{\sigma,\eta}(\lambda)\begin{cases} = \lambda & \lambda \geq \sigma(1+\eta) \\ = 0 & \lambda < \sigma(1-\eta) \\ \in [0, \lambda] & \text{otherwise} \end{cases} \tag{10}$$

Note that $P_{\sigma,\eta}$ is not fully specified in the range $[\sigma(1-\eta), \sigma(1+\eta))$, so $A_{\sigma,\eta}$ is any of a family of matrices with error η.

For intuition, observe that $\|P_{\sigma,\eta}^{(\mathrm{SV})}(A)V(\cdot, i)\| = 0$ for $i > \ell_A(\sigma(1-\eta))$, $\|(A - P_{\sigma,\eta}^{(\mathrm{SV})}(A))V(\cdot, i)\| = 0$ for $i \leq \ell_A(\sigma(1+\eta))$, and $\|P_{\sigma,\eta}^{(\mathrm{SV})}(A)V(\cdot, i)\| \leq \|AV(\cdot, i)\|$ for $i \in (\ell_A(\sigma(1+\eta)), \ell_A(\sigma(1-\eta))]$. That is, $P_{\sigma,\eta}^{(\mathrm{SV})}(A)$ preserves singular vectors with value $\geq \sigma(1+\eta)$, sends those with value $< \sigma(1-\eta)$ to zero, and does something in between for the rest of the singular vectors.

2.2 Sample and query access oracles

Since our algorithms run in time sublinear in input size, we will take care in describing how we need to access our input. Conveniently, the sample and query oracle we present below will be a good classical analogue to a quantum state, and will be used heavily to move between intermediate steps of these quantum-inspired algorithms. First, a simple query oracle allowing some error:

Definition 2.5 (Query access)**.** For a vector $v \in \mathbb{C}^n$, we have $\mathrm{Q}(v)$, *query access* to v if for all $i \in [n]$, we can obtain v_i. Likewise, for a matrix $A \in \mathbb{C}^{m \times n}$, we have $\mathrm{Q}(A)$ if for all $(i, j) \in [m] \times [n]$, we can obtain $A(i, j)$.

For example, in the typical RAM access model, we are given our input $v \in \mathbb{C}^n$ as $\mathrm{Q}(v)$.

Definition 2.6 (Sample and query access of a vector)**.** For a vector $v \in \mathbb{C}^n$, we have $\mathrm{SQ}_\nu(v)$, *sample and query access* to v with norm error ν, if we can:

1. Perform independent samples $i \in [n]$ following the distribution $\mathcal{D}_v$, where $\mathcal{D}_v(i) = |v(i)|^2/\|v\|^2$ with (expected) cost $\mathbf{s}(v)$;
2. Query as in $\mathrm{Q}(v)$ with (expected) cost $\mathbf{q}(v)$;
3. Obtain $\|v\|$ up to ν multiplicative error with success probability at least $9/10$ in (expected) cost $\mathbf{n}_\nu(v)$.

We define $\mathbf{sq}_\nu(v) := \mathbf{s}(v) + \mathbf{q}(v) + \mathbf{n}_\nu(v)$.[ii]

Definition 2.7 (Sample and query access of a matrix)**.** For a matrix $A \in \mathbb{C}^{m \times n}$, we have $\mathrm{SQ}_{\nu_1}^{\nu_2}(A)$ if:

1. We have $\mathrm{SQ}_{\nu_1}(A(i, \cdot))$ for all $i \in [m]$ with (expected) sample, query, and norm estimation complexity bounds denoted by $\mathbf{s}(A)$, $\mathbf{q}(A)$ and $\mathbf{n}_{\nu_1}(A)$ respectively;

[ii] For avoiding unnecessary complications we assume that $\mathbf{s}(v) \geq 1, \mathbf{q}(v) \geq 1$, and $\mathbf{n}_\nu(v) \geq 1$.

2. We can sample from $\mathcal{D}_a$ with complexity $\mathbf{s}(A)$ for $a \in \mathbb{R}^m$ the vector of row norms, i.e., $a_i = \left\|A(i,\cdot)\right\|_2$;
3. We can estimate $\|A\|_F$ to multiplicative error ν_2 in complexity $\mathbf{n}^{\nu_2}(A)$.

We define the cost $\mathbf{sq}_{\nu_1}^{\nu_2}(A) := \mathbf{s}(A) + \mathbf{q}(A) + \mathbf{n}_{\nu_1}(A) + \mathbf{n}^{\nu_2}(A)$. In case ν_1 or ν_2 is zero, we simply omit the corresponding sub/superscripts. When we work with matrices $A^{(1)}, \ldots, A^{(\tau)}$, we use the notation $\mathbf{sq}_{\nu_1}^{\nu_2}(A^{[\tau]}) := \mathbf{sq}_{\nu_1}^{\nu_2}(A^{(1)}, \ldots, A^{(\tau)}) := \max_{\ell\in[\tau]} \mathbf{sq}_{\nu_1}^{\nu_2}(A^{(i)})$. Similarly we define $\mathbf{sq}_{\nu_1}^{\nu_2}(A^{(\dagger)}) := \max\{\mathbf{sq}_{\nu_1}^{\nu_2}(A), \mathbf{sq}_{\nu_1}^{\nu_2}(A^\dagger)\}$.

With abuse of notations, we do not distinguish $\mathrm{SQ}(U)$ and $\mathrm{SQ}(U^\dagger)$ if the context is clear. Given $\mathrm{Q}(v)$, we also have $\mathrm{SQ}(v)$ in $O(n)$ time per sample. Quantum-inspired algorithms generally achieve exponential speedups when each sample takes time and query complexity $\mathrm{poly}(\log n)$. The definition of matrix sampling has been given earlier by [DKR02, FKV04]. With this definition, $\mathrm{SQ}(A)$ implies $\mathrm{SQ}(\mathrm{vec}(A))$: sampling can be done by first sampling $i \sim a$ (for a the vector of row norms of A), and then sampling $j \sim A(i,\cdot)$. This definition can be made slightly more general by adding other forms of error, but in the settings we care about, we can translate them into more useful and familiar types of error. For brevity we will sometimes abuse notation and write "$\mathrm{Q}(v) \in \mathbb{C}^n$" instead of "$\mathrm{Q}(v)$ for $v \in \mathbb{C}^n$", and similarly "$\mathrm{SQ}(A) \in \mathbb{C}^{m\times n}$" instead of "$\mathrm{SQ}(A)$ for $A \in \mathbb{C}^{m\times n}$".

Remark 2.8. We can get sample and query access to input matrices and vectors in the following settings:

- (Uniformity) Given $\mathrm{Q}(v) \in \mathbb{C}^n$, if we know some $C = c\|v\|$ such that $\max |v_i| \le \frac{C}{\sqrt{n}}$, then we can get $\mathrm{SQ}_\nu(v)$ where $\mathbf{s}(v) = \mathcal{O}\big(c^2 \log\frac{1}{\delta}\mathbf{q}(v)\big)$ time and $\mathbf{n}(v) = \mathcal{O}\big(c^2\frac{1}{\nu^2}\log\frac{1}{\delta}\mathbf{q}(v)\big)$. In this model, a quantum state corresponding to v can be prepared in time $\mathcal{O}(\log n)$.

- (Weak sample access) Given $\mathrm{Q}(v) \in \mathbb{C}^n$ and access to probabilities and samples from a distribution $\mathcal{D}$ such that $\mathcal{D}(i) \ge c^2\frac{|v_i|^2}{\|v\|^2}$, we have $\mathrm{SQ}_\nu(v)$ where $\mathbf{s}(v) = \mathcal{O}\big(c^2 \log\frac{1}{\delta}\mathbf{q}(v)\big)$ time and $\mathbf{n}(v) = \mathcal{O}\big(c^2\frac{1}{\nu^2}\log\frac{1}{\delta}\mathbf{q}(v)\big)$.

- (Partial sums) For $v \in \mathbb{C}^n$, if we can compute in $\mathcal{O}(T)$ time $\sum |v_i|^2$ when i iterates over the set of indices that, in binary, begins with a particular bit string, then we have $\mathrm{SQ}(v)$ where $\mathbf{q}(v) = \mathcal{O}(T), \mathbf{s}(v) = \mathcal{O}(T\log n), \mathbf{n}(v) = \mathcal{O}(T)$. In this model, a quantum state corresponding to v can be prepared in time $\tilde{\mathcal{O}}(T)$ by the method in [GR02].

- (Sparsity) If $A \in \mathbb{C}^{m\times n}$ has at most s non-zero entries per row, and $|A(i,j)| \le c$, then we can sample from $A(i,\cdot)$ in time $\mathcal{O}(s)$ for all $i \in [m]$. In this model, a quantum state corresponding to v can be prepared efficiently using $\mathcal{O}(s)$ operations.

- (Dynamic data structure) If $A \in \mathbb{C}^{m\times n}$ is stored in a data structure which supports direct norm and query access as in [CLLW19, Tan19, CLW18, GLT18] (see also Fig. 1 for an example), then we can sample from $A(i,\cdot)$ in time $\mathcal{O}\big(\log^2(mn)\big)$ for all $i \in [m]$. If this data structure is implemented in the QRAM model, we can also prepare the corresponding quantum state in time $\mathrm{poly}(\log(mn))$ as in [KP17].

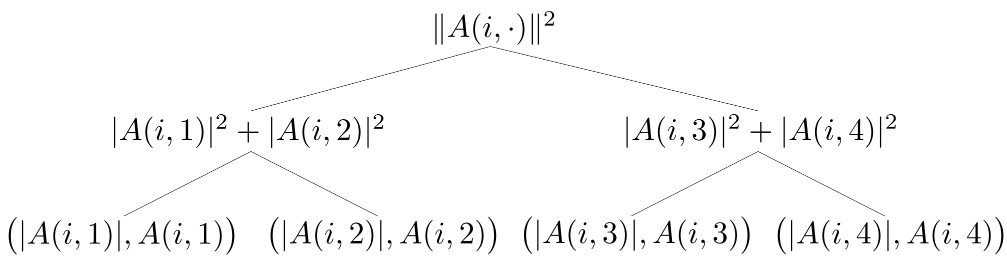

Figure 1: Illustration of a data structure that allows for sampling access to a row of $A \in \mathbb{C}^{4\times 4}$. The leaf node contains both the absolute value as well as the original value of an entry.

3 Main results and technical tools

3.1 Main results

We consider a function $f : [0,\infty) \to \mathbb{C}$ that is L-Lipschitz continuous, i.e., $|f(x_1)-f(x_2)| \le L|x_1-x_2|$ for all $x_1, x_2 \in [0,\infty)$, and maps 0 to 0. One of the main results of this paper it to develop methods to obtain SQ access to a matrix that is close to $f^{(\mathrm{SV})}(A)$ for some matrix A. The result is as follows. (We assume for simplicity that function evaluation takes $\mathcal{O}(1)$ arithmetic operations.)

Theorem 3.1 (Singular value transformation)**.** *Let $A = A^{(1)} + \cdots + A^{(\tau)} \in \mathbb{C}^{m\times n}$. For $\epsilon \in (0, L)$, $\nu = \widetilde{\mathcal{O}}\big(\epsilon^2/(\alpha^2 \sum_\ell \|A^{(\ell)}\|_F^2)\big)$ there exists an algorithm with time complexity $\widetilde{\mathcal{O}}\bigg(\Big(\frac{\tau L^2(\sum_\ell \|A^{(\ell)}\|_F^2)}{\epsilon^2}\Big)^{18}\mathbf{sq}(A^{[\tau]\,(\dagger)})\bigg)$ which provides succinct descriptions of $\widetilde{\mathcal{O}}\Big(\frac{\epsilon^2}{L^2\|A\|_F^2}\Big)$-approximate isometries $\check{V}$, $\check{U}$, and a diagonal matrix $D \in \mathbb{R}^{r\times r}$ such that the matrix $B := \check{U}D\check{V}^\dagger$ satisfies $\big\|B - f^{(\mathrm{SV})}(A)\big\| \le \epsilon$ with probability $9/10$. Moreover, the provided representation of B enables $\mathrm{SQ}_\nu^\nu(B)$ access in time $\mathbf{s}(B) = \mathcal{O}\big(r^2\mathbf{sq}(\check{U},\check{V})\big)$, $\mathbf{q}(B) = \mathcal{O}\big(r\mathbf{sq}(\check{U},\check{V})\big)$, $\mathbf{n}_\nu(B) = \mathbf{n}^\nu(B) = \mathcal{O}\big(r^2/\nu^2\mathbf{sq}(\check{U},\check{V})\big)$, where $r \le \ell_{\frac{A}{\|A\|_F}}(\widetilde{\Theta}(\frac{\epsilon}{L\|A\|_F})) = \widetilde{\mathcal{O}}\Big(\frac{L^2\|A\|_F^2}{\epsilon^2}\Big)$ and $\mathbf{sq}(\check{U},\check{V}) = \widetilde{\mathcal{O}}\bigg(\Big(\frac{\tau L^2(\sum_\ell \|A^{(\ell)}\|_F^2)}{\epsilon^2}\Big)^{13}\mathbf{sq}(A^{[\tau]\,(\dagger)})\bigg)$.*

The proof of this theorem is in Section 6.1.

Remark 3.2. When f is an even function, then "even" singular value transformation [GSLW19] can be performed more efficiently by using techniques derived from [KV17]. Let $A = \sum_{i=1}^n \sigma_i u_i v_i^\dagger \in \mathbb{C}^{m\times n}$ be a singular value decomposition, then we define the "even" singular value transform of A as $f^{(\mathrm{even})}(A) := \sum_{i=1}^n f(\sigma_i) v_i v_i^\dagger$. For an even function f we can define $g(x) := f(\sqrt{x})$ so that $f^{(\mathrm{even})}(A) = g(A^\dagger A)$, and we can apply Corollary B.4 with g.

The result for singular value transformation naturally extends to a more special case, namely, functions for Hermitian matrices. Now, we consider a function $f : \mathbb{R} \to \mathbb{C}$ that is L-Lipschitz continuous, i.e., $|f(x_1) - f(x_2)| \le L|x_1 - x_2|$ for all $x_1, x_2 \in \mathbb{R}$. We also develop methods to sample and query to a matrix that is close to $f(A)$ for some Hermitian matrix A. The result is as follows.

Theorem 3.3 (Matrix function)**.** *Let $A = \gamma I + A^{(1)} + \cdots + A^{(\tau)}$ where for each $A^{(\ell)} \in \mathbb{C}^{n\times n}$ is Hermitian. For $\epsilon \in (0, L)$, $\nu = \widetilde{\mathcal{O}}\big(\epsilon^2/(L^2 \sum_\ell \|A^{(\ell)}\|_F^2)\big)$ there exists an algorithm with time complexity $\widetilde{\mathcal{O}}\bigg(\Big(\frac{\tau L^2(\sum_\ell \|A^{(\ell)}\|_F^2)}{\epsilon^2}\Big)^{18}\mathbf{sq}(A^{[\tau]})\bigg)$ which provides a succinct description of a $\widetilde{\mathcal{O}}\Big(\frac{\epsilon^2}{L^2\|A\|_F^2}\Big)$-approximate isometry $\check{U}$, and a diagonal matrix $D \in \mathbb{R}^{r\times r}$ such that the matrix $B := \check{U}D\check{U}^\dagger$ satisfies*

$\|f(\gamma)I + B - f(A)\| \leq \epsilon$ *with probability* $9/10$*. Moreover, the provided representation of* B *enables* $\mathrm{SQ}_\nu^\nu(B)$ *access in time* $\mathbf{s}(B) = \mathcal{O}\big(r^2\mathbf{sq}(\check{U})\big)$, $\mathbf{q}(B) = \mathcal{O}\big(r\mathbf{sq}(\check{U})\big)$, $\mathbf{n}_\nu(B) = \mathbf{n}^\nu(B) = \mathcal{O}\big(r^2/\nu^2\mathbf{sq}(\check{U})\big)$, *where* $r \leq \ell_{\frac{A}{\|A\|_F}}(\widetilde{\Theta}(\frac{\epsilon}{L\|A\|_F})) = \widetilde{\mathcal{O}}\Big(\frac{L^2\|A\|_F^2}{\epsilon^2}\Big)$ *and* $\mathbf{sq}(\check{U}) = \widetilde{\mathcal{O}}\bigg(\Big(\frac{\tau L^2(\sum_\ell \|A^{(\ell)}\|_F^2)}{\epsilon^2}\Big)^{13}\mathbf{sq}(A^{[\tau]})\bigg)$*.*

The proof of this theorem is in Section 6.2.

To prove the main theorems as well as the results in our applications, we need some technical tools that were developed or derived from previous results. We summarize them in the next subsection.

3.2 Technical tools

The first tool concerns estimating the inner product of two vectors.

Proposition 3.4 (Inner product estimation, Proposition 4.2 of [Tan19])**.** *Given* $\mathrm{SQ}(u), \mathrm{Q}(v) \in \mathbb{C}^n$*, we can estimate* $\langle u, v\rangle$ *to* ϵ *additive error and failure probability* δ *in query and time complexity* $\mathcal{O}\big(\|u\|^2\|v\|^2\frac{1}{\epsilon^2}\log\frac{1}{\delta}(\mathbf{s}(u) + \mathbf{q}(u) + \mathbf{q}(v)) + \mathbf{n}(u)\big)$*.*

Proof. Define a random variable Z as follows:

$$Z = \frac{v(i)\|u\|^2}{u(i)} \text{ with probability } \frac{\|u(i)\|^2}{\|u\|^2}. \tag{11}$$

The expected value and variance of Z are

$$\mathbb{E}[Z] = \langle u, v\rangle \quad \text{and} \quad \mathrm{Var}[|Z|] \leq \sum_i \frac{|v(i)|^2\|u\|^4}{|u(i)|^2}\frac{|u(i)^2|}{\|u\|^2} = \|u\|^2\|v\|^2. \tag{12}$$

Then, we prove this lemma by the technique of median of means. Given pq (where p and q will be determined later) samples of Z's, we divide these samples into p groups. Let $Y_i = \frac{\sum_{j=1}^q Z_j}{p}$ be the mean of the i-th group and $\tilde{Y}$ be the median of $\{Y_1, \ldots, Y_p\}$. The observation is that the median $\tilde{Y}$ is greater than $\mathbb{E}[Z] + \epsilon$ if and only if more than $p/2$ of means in $\{Y_1, \ldots, Y_p\}$ are greater than $\mathbb{E}[Z] + \epsilon$.

First, we show the probability that Y_i is much larger than $\mathbb{E}[Z]$ is bounded for all i. We use the Chebyshev inequality for complex random variables as

$$\Pr[|Y_i - \mathbb{E}[Z]| \geq \epsilon] \leq \frac{\mathrm{Var}[|Z|]}{\epsilon^2 q}. \tag{13}$$

Let $q = \frac{4\mathrm{Var}[|Z|]}{\epsilon^2}$ so that the above probability is at most $1/4$. Then, let E_i be the event that $Y_i - \mathbb{E}[Z] > \epsilon$ for $i \in \{1, \ldots, p\}$. By the Chernoff-Hoeffding inequality, we have

$$\Pr\left[\sum_{i=1}^p E_i - p\Pr[E_i] \geq p/4\right] \leq e^{-p/8}. \tag{14}$$

The probability that the event $\tilde{Y} - \mathbb{E}[Z] > \epsilon$ happens is bounded by

$$\begin{aligned}
\Pr[|\tilde{Y} - \mathbb{E}[Z]| \leq \epsilon] &\leq \Pr[\tilde{Y} - \mathbb{E}[Z] \leq \epsilon] && (15)\\
&= 1 - \Pr[\tilde{Y} - \mathbb{E}[Z] \geq \epsilon] && (16)\\
&= 1 - \Pr\left[\sum_{i=1}^{p} E_i \geq p/2\right] && (17)\\
&\leq \Pr\left[\sum_{i=1}^{p} E_i - p\Pr[E_i] \geq p/4\right] && (18)\\
&\leq 1 - e^{-p/8}. && (19)
\end{aligned}$$

Let $\delta = e^{-p/8}$. By sampling X for a number of $pq = O(\|u\|^2\|v\|^2 \frac{1}{\epsilon^2}\log(\frac{1}{\delta}))$ times, dividing them randomly in p groups, and outputting the median of means of these groups, one obtains an estimate of $x^\dagger Ay$ with additive error at most ϵ and success probability $1-\delta$. The time cost of one sample of Z is $\mathbf{s}(u) + \mathbf{q}(u) + \mathbf{q}(v)$, and we need to know the norm of u to calculate Z, so the total time complexity is $\mathcal{O}\Big(\frac{\|v\|^2\|u\|^2}{\epsilon^2}\log\frac{1}{\delta}(\mathbf{s}(u) + \mathbf{q}(u) + \mathbf{q}(v)) + \mathbf{n}(u)\Big)$. □

Remark 3.5. Note that this inner product procedure can be used for higher-order tensors as well:

(a) (Trace inner products, Lemma 11 of [GLT18]) Given $\mathrm{SQ}(A) \in \mathbb{C}^{n\times n}$ and $\mathrm{Q}(B) \in \mathbb{C}^{n\times n}$, we can estimate $\mathrm{Tr}[AB]$ to additive error ϵ with probability at least $1-\delta$ by using

$$\mathcal{O}\bigg(\frac{\|A\|_F^2\|B\|_F^2}{\epsilon^2}\big(\mathbf{s}(A) + \mathbf{q}(A) + \mathbf{q}(B)\big)\log\frac{1}{\delta} + \mathbf{n}(A)\bigg) \tag{20}$$

time and queries.

To do this, note that $\mathrm{SQ}(A)$ and $\mathrm{Q}(B)$ imply $\mathrm{SQ}(\mathrm{vec}(A))$ and $\mathrm{Q}(\mathrm{vec}(B^\dagger))$. $\mathrm{Tr}[AB] = \langle \mathrm{vec}(B^\dagger), \mathrm{vec}(A)\rangle$, so we can just apply Proposition 3.4 to conclude.

(b) (Expectation values) Given $\mathrm{SQ}(A) \in \mathbb{C}^{n\times n}$ and $\mathrm{Q}(x), \mathrm{Q}(y) \in \mathbb{C}^n$, we can estimate $x^T Ay$ to additive error ϵ with probability at least $1-\delta$ by using

$$\mathcal{O}\bigg(\frac{\|A\|_F^2\|x\|^2\|y\|^2}{\epsilon^2}\big(\mathbf{s}(A) + \mathbf{q}(A) + \mathbf{q}(x) + \mathbf{q}(y)\big)\log\frac{1}{\delta} + \mathbf{n}(A)\bigg) \tag{21}$$

time and queries.

To do this, observe that $x^T Ay = \mathrm{Tr}(x^T Ay) = \mathrm{Tr}(Ayx^T)$ and that $\mathrm{Q}(yx^T)$ can be simulated with $\mathrm{Q}(x), \mathrm{Q}(y)$. So, we just apply the trace inner product procedure.

Let $V \in \mathbb{C}^{n\times k}$ and $w \in \mathbb{C}^n$. The following proposition shows that we can sample and query Vw with bounded error.

Proposition 3.6 (Small linear combinations, Proposition 4.3 of [Tan19])**.** *Given* $\mathrm{SQ}(V^T) \in \mathbb{C}^{k\times n}$ *and* $\mathrm{Q}(w) \in \mathbb{C}^k$, *we can get* $\mathrm{SQ}_\nu(Vw)$ *with (expected) time complexities*

$$
\begin{aligned}
\mathbf{q}(Vw) &= \mathcal{O}(k(\mathbf{q}(V)+\mathbf{q}(w)), && (22)\\
\mathbf{s}(Vw) &= \mathcal{O}(k\mathsf{C}_{V,w}(\mathbf{s}(V)+k\mathbf{q}(V)+k\mathbf{q}(w))) && (23)\\
\mathbf{n}_\nu(Vw) &= \mathcal{O}\bigg(\frac{k\mathsf{C}_{V,w}}{\nu^2}\log\frac{1}{\delta}(\mathbf{s}(V)+k\mathbf{q}(V))\bigg), && (24)
\end{aligned}
$$

with success probability at least $1-\delta$ *and* $\mathsf{C}_{V,w} = \big(\sum_i |w(i)|^2\|V(\cdot,i)\|^2\big)/\|\sum_i w(i)V(\cdot,i)\|^2$.

Proof. We get the first expression by explicitly calculating $[Vw](i) = \sum_{j=1}^r V(i,j)w(j)$. To sample from the vector Vw, we use the rejection sampling as in [Tan19]. We set the initial distribution P and the target distribution Q to be

$$
\begin{aligned}
P(x) &:= \sum_{\ell=1}^k \frac{|V(\cdot,\ell)w(\ell)|^2}{\sum_{j=1}^k |V(\cdot,\ell)w(j)|^2}\,\frac{|V(x,\ell)|^2}{\|V(\cdot,\ell)\|^2} && (25)\\
Q(x) &:= \frac{|\sum_{j=1}^k V(x,j)w(j)|^2}{\|\sum_{j=1}^k V(\cdot,j)w(j)\|^2} && (26)
\end{aligned}
$$

for $x \in [n]$.

$$
\frac{Q(x)}{P(x)} = \frac{|\sum_{\ell=1}^k V(x,\ell)w(\ell)|^2}{\sum_{\ell=1}^k |V(x,\ell)|^2|w(\ell)|^2}\,\frac{\sum_{j=1}^k \|V(\cdot,j)\|^2|w(j)|^2}{\|\sum_{j=1}^k V(\cdot,j)w(j)\|^2} \le k\mathsf{C}_{V,w}, \tag{27}
$$

where the inequality follows from Cauchy-Schwarz. We set the probability that we output a sample on x as

$$
R(x) := \frac{|\sum_{j=1}^k V(x,j)w(j)|^2}{k\cdot\sum_{\ell=1}^k |V(x,\ell)|^2|w(\ell)|^2}. \tag{28}
$$

Note that we can compute $R(x)$ exactly in time $k(\mathbf{q}(V)+\mathbf{q}(w))$. Hence, we can sample from Vw in expected running time $\mathcal{O}(k\mathsf{C}_{V,w}(\mathbf{s}(V)+k\mathbf{q}(V)+k\mathbf{q}(w)))$.

To estimate the norm of Vw, it suffices to estimate the probability the rejection sampling succeeds. The probability that the rejection sampling succeeds is

$$
\sum_{x=1}^n P(x)R(x) = \frac{\sum_{x=1}^n |\sum_{j=1}^k V(x,j)w(j)|}{k\sum_{\ell=1}^k \|V(\cdot,\ell)\|^2|w(\ell)|^2} = \frac{\|Vw\|^2}{k\sum_{\ell=1}^k \|V(\cdot,\ell)\|^2|w(\ell)|^2} = \frac{1}{k\mathsf{C}_{V,w}}. \tag{29}
$$

We repeat the rejection sampling p times (which we will specify soon) and let X be the number of successful trials. Then, by the Chernoff-Hoeffding inequality,

$$
\Pr\bigg[\Big|\frac{X}{p}-\frac{1}{k\mathsf{C}_{V,w}}\Big| \le \frac{\nu}{k\mathsf{C}_{V,w}}\bigg] = \Pr\bigg[\Big|X-\frac{p}{k\mathsf{C}_{V,w}}\Big| \le \frac{p\nu}{k\mathsf{C}_{V,w}}\bigg] \ge 1-2e^{-\frac{\nu^2 p}{2k\mathsf{C}_{V,w}}} = 1-\delta. \tag{30}
$$

Therefore, by repeating the trials $p = \mathcal{O}\big(k\mathsf{C}_{V,w}\frac{1}{\nu^2}\log\frac{1}{\delta}\big)$ times, we can estimate $p = \frac{1}{k\mathsf{C}_{V,w}}$ with multiplicative error ν with probability at least $1-\delta$. Finally, since we can compute $k\sum_{\ell=1}^k \|V(\cdot,\ell)\|^2|w(\ell)|^2$ exactly, we can obtain $\|Vw\|$ with the desired error from our estimation of $\frac{1}{k\mathsf{C}_{V,w}}$.

□

With the following proposition, we can simulate the sample-query access of Ax given a succinct description of A.

Proposition 3.7. *Let $S, T \in \mathcal{C}^{k\times n}$, $u_1, \ldots, u_r, v_1, \ldots, v_r \in \mathcal{C}^k$, and $\sigma_1, \ldots, \sigma_r, \rho_1, \ldots, \rho_r \in \mathcal{C}$. Let $U, V \in \mathcal{C}^{n\times r}$ be α-approximate isometries*[iii] *such that $U(\cdot, j) = \frac{S^\dagger u_j}{\sigma_j}$ and $V(\cdot, j) = \frac{T^\dagger v_j}{\rho_j}$ for $j \in [r]$. Let $D \in \mathbb{C}^{r\times r}$ be a diagonal matrix and $x \in \mathbb{C}^n$ be a vector. Then, given $\mathrm{SQ}(S)$, $\mathrm{SQ}(T)$, $\mathrm{Q}(u_i)$ and $\mathrm{Q}(v_i)$ for $i \in [r]$, σ_i and ρ_i for $i \in [r]$, $\mathrm{Q}(D)$, and $\mathrm{Q}(x)$, we can obtain $\mathrm{SQ}_\nu(x')$ of an x' such that $\|(UDV^\dagger)x - x'\| \le \epsilon$ The complexities of query and sample access to x' are*

$$\mathbf{q}(x') = \mathcal{O}\left(r(\mathbf{q}(U) + \mathbf{q}(D)) + \frac{r^3\|x\|^2\|D\|^2(1+\alpha)^4}{\epsilon^2}\log\frac{r}{\delta}(\mathbf{s}(V) + \mathbf{q}(V) + \mathbf{q}(x))\right) \tag{31}$$

$$\begin{aligned}\mathbf{s}(x') = \mathcal{O}\bigg(& r\frac{(1+\alpha)^2}{(1-\alpha)}[\mathbf{s}(U) + r(\mathbf{q}(U) + \mathbf{q}(D)) \\ &+ \frac{r^3\|x\|^2\|D\|^2(1+\alpha)^4}{\epsilon^2}\log\left(\frac{r^2(1+\alpha)^2}{\delta(1-\alpha)}\right)(\mathbf{s}(V) + \mathbf{q}(V) + \mathbf{q}(x))]\bigg).\end{aligned} \tag{32}$$

Here,

$$\mathbf{s}(V) := \max_j \mathcal{O}\big(k\mathsf{C}_{T,v_j}(\mathbf{s}(T) + k\mathbf{q}(v_j) + k\mathbf{q}(T))\big) \qquad \mathbf{q}(V) := \max_j \mathcal{O}(k(\mathbf{q}(T) + \mathbf{q}(v_j))) \tag{33}$$

$$\mathbf{s}(U) := \max_j \mathcal{O}\big(k\mathsf{C}_{S,u_j}(\mathbf{s}(S) + k\mathbf{q}(u_j) + k\mathbf{q}(S))\big) \qquad \mathbf{q}(U) := \max_j \mathcal{O}(k(\mathbf{q}(S) + \mathbf{q}(u_j))). \tag{34}$$

Proof. By Proposition 3.6, we can obtain the query and sample accesses to $V(\cdot, j), U(\cdot, j)$ for $j \in [r]$ in expected running time

$$\mathbf{q}(V(\cdot, j)) = \mathcal{O}(k\mathbf{q}(T) + k\mathbf{q}(v_j)) \qquad \mathbf{s}(V(\cdot, j)) = \mathcal{O}\big(k\mathsf{C}_{T,v_j}(\mathbf{s}(T) + k\mathbf{q}(v_j) + k\mathbf{q}(T))\big) \tag{35}$$

$$\mathbf{q}(U(\cdot, j)) = \mathcal{O}(k\mathbf{q}(S) + k\mathbf{q}(u_j)) \qquad \mathbf{s}(U(\cdot, j)) = \mathcal{O}\big(k\mathsf{C}_{S,u_j}(\mathbf{s}(S) + k\mathbf{q}(u_j) + k\mathbf{q}(S))\big). \tag{36}$$

Consider the column vector $(V^\dagger x) \in \mathbb{C}^r$. we want to approximate it with a vector w. By Proposition 3.4, we can give an estimate of $[V^\dagger x](j)$ with additive error $\frac{\epsilon}{\sqrt{r}\|D\|(1+\alpha)}$ with probability at least $1-\delta$ in time

$$\mathcal{O}\left(\frac{r\|x\|^2\|V(\cdot, i)\|^2\|D\|^2(1+\alpha)^2}{\epsilon^2}\log\frac{1}{\delta}(\mathbf{s}(V) + \mathbf{q}(V) + \mathbf{q}(x))\right).$$

Therefore, if we can set an vector $w(i)$ as the above estimate of $(V^\dagger x)$, we have

$$\left\|V^\dagger x - w\right\|_F \le \frac{\epsilon}{\|D\|(1+\alpha)} \tag{37}$$

$$\mathbf{q}(w) = \mathcal{O}\left(\frac{r\|x\|^2\|V\|_F^2\|D\|^2(1+\alpha)^2}{\epsilon^2}\log\frac{1}{\delta}(\mathbf{s}(V) + \mathbf{q}(V) + \mathbf{q}(x))\right), \tag{38}$$

with success probability at least $1-\delta$.

Note that $x' := UDw$ is sufficiently close to $UDV^\dagger x$:

$$\begin{aligned}\|(UDV^\dagger)x - x'\| &= \|UD(V^\dagger x) - UDw\| \le \|UD\|\|V^\dagger x - w\|_F \\ &\le \|U\|\|D\|\frac{\epsilon}{\|D\|(1+\alpha)} \le \epsilon.\end{aligned} \tag{39}$$

[iii] As defined in Definition 2.1

The last inequality follows from the fact that U is α-approximate orthogonal (Remark 2.2). Therefore, It suffices to give $\mathrm{SQ}_\nu(UDw)$.

To get $\mathrm{SQ}(UDw)$, we first give queries to Dw, which requires r queries to D and w and so it has running time

$$\mathbf{q}(Dw) = r\mathbf{q}(D) + r\mathbf{q}(w). \tag{40}$$

Next, we obtain sampling and query accesses to $U(Dw)$ by Proposition 3.6, which requires the running time

$$\mathbf{q}(U(Dw)) = \mathcal{O}(r(\mathbf{q}(U) + \mathbf{q}(Dw))), \tag{41}$$
$$\mathbf{s}(U(Dw)) = \mathcal{O}(r\mathsf{C}_{U,Dw}(\mathbf{s}(U) + r\mathbf{q}(U) + r\mathbf{q}(Dw))). \tag{42}$$

Since U is an α-approximate isometry, we have

$$\mathsf{C}_{U,Dw} = \frac{\sum_i |[Dw](i)|^2 \|U(\cdot,i)\|^2}{\|\sum_i [Dw](i)U(\cdot,i)\|^2} = \frac{\sum_i |[Dw](i)|^2 \|U(\cdot,i)\|^2}{\|UDw\|^2} \leq \frac{\|[Dw]\|^2(1+\alpha)^2}{\|Dw\|^2(1-\alpha)} = \frac{(1+\alpha)^2}{(1-\alpha)}, \tag{43}$$

where we used that $\|UDw\|^2 \geq \|U_0Dw\|^2 - \|(U-U_0)Dw\|^2 \geq (1-\alpha)\|Dw\|^2$, where U_0 is an isometry. Also note that $\|V\|_F^2 \leq r(1+\alpha)^2$. Since we are calling $\mathbf{q}(w)$ multiple times, we need to have it's error probability to be 1/(number of times called). Putting everything together, we have

$$\begin{aligned}
\mathbf{q}(U(Dw)) &= \mathcal{O}(r\mathbf{q}(U) + r\mathbf{q}(D) + r\mathbf{q}(w;\delta/r)) \\
&= \mathcal{O}\left(r(\mathbf{q}(U) + \mathbf{q}(D)) + \frac{r^3\|x\|^2\|D\|^2(1+\alpha)^4}{\epsilon^2}\log\frac{r}{\delta}(\mathbf{s}(V) + \mathbf{q}(V) + \mathbf{q}(x))\right) \qquad (44)\\
\mathbf{s}(U(Dw)) &= \mathcal{O}\left(r\frac{(1+\alpha)^2}{(1-\alpha)}(\mathbf{s}(U) + r\mathbf{q}(U) + r\mathbf{q}(D) + r\mathbf{q}(w;\delta\frac{(1-\alpha)}{r^2(1+\alpha)^2}))\right), \text{and} \\
&= \mathcal{O}\left(r\frac{(1+\alpha)^2}{(1-\alpha)}[\mathbf{s}(U) + r(\mathbf{q}(U) + \mathbf{q}(D))\right. \\
&\quad \left. + \frac{r^3\|x\|^2\|D\|^2(1+\alpha)^4}{\epsilon^2}\log\left(\frac{r^2(1+\alpha)^2}{\delta(1-\alpha)}\right)(\mathbf{s}(V) + \mathbf{q}(V) + \mathbf{q}(x))]\right). \qquad (45)
\end{aligned}$$

□

If one cannot compute a desired $\mathrm{SQ}(x) \in \mathbb{C}^n$, but can instead compute some $\mathrm{SQ}(x') \in \mathbb{C}^n$ such that $\|x' - x\| \leq \epsilon$, then the following lemma can be used to bound the total variation distance between sampling distributions corresponding to x and x':

Lemma 3.8 ([Tan19])**.** *Given $x, y \in \mathbb{C}^n$ such that $\|x - y\| \leq \epsilon$, we have $\left\|\mathcal{D}_x - \mathcal{D}_y\right\|_{TV} \leq 2\epsilon/\|x\|$.*

The following theorem is an extension of [KV17], which says that given efficient $\mathrm{SQ}(A^\dagger)$, $\mathrm{SQ}(B)$ access for matrices A and B, we get efficient (approximate) $\mathrm{SQ}(AB)$ access. This result can also be generalized to the case when $A = A^{(1)} + A^{(2)} + \ldots + A^{(\tau)}$ and/or $B = B_1 + B_2 + \ldots + B_\tau$, but for simplicity we only discuss the former case.

Theorem 3.9 (Matrix multiplication)**.** *Let $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{n\times p}$, and $\epsilon, \delta \in (0,1)$ be error parameters. We can get a concise description of $U \in \mathbb{C}^{m\times r}, D \in \mathbb{R}^{r\times r}, V \in \mathbb{C}^{r\times p}$ such that with 0.9 probability $\|AB - UDV\|_F \leq \epsilon$, moreover $r \leq \mathcal{O}\Big(\|A\|_F^2\|B\|_F^2/\epsilon^2\Big)$, $\big\|U^\dagger U - I\big\| \leq \delta$, and $\big\|VV^\dagger - I\big\| \leq \delta$. The complexities are $\mathbf{q}(U) = \mathcal{O}\big(t\mathbf{q}(A^\dagger)\big)$, $\mathbf{s}(U) = \mathcal{O}\big(t(\mathbf{s}(A^\dagger) + t\mathbf{q}(A^\dagger))\big)$, for $t = \mathcal{O}\Big(\|A\|_F^2\|B\|_F^2/\epsilon^2\Big)$, and $\mathbf{n}_{\mathcal{O}(\delta)}(U) = \mathcal{O}(1)$. Similarly $\mathbf{q}(V) = \mathcal{O}(t\mathbf{q}(B))$, $\mathbf{s}(V) = \mathcal{O}(t(\mathbf{s}(B) + t\mathbf{q}(B)))$, for some $t = \mathcal{O}\Big(\|A\|_F^2\|B\|_F^2/\epsilon^2\Big)$, and $\mathbf{n}_{\mathcal{O}(\delta)}(V) = \mathcal{O}(1)$. Finally, computing the succinct description takes time $\mathcal{O}\Big(t^3 + \frac{t^2}{\delta^2}(\mathbf{sq}(A^\dagger) + \mathbf{sq}(B))\Big)$.*

Proof. Let us assume without loss of generality[iv] that $\|A\|_F = \|B\|_F$. Inspired by [KV17, Theorem 2.1], we sample $s = \mathcal{O}\Big(\|A\|_F^2\|B\|_F^2/\epsilon^2\Big)$ indices $i_1, i_2, \ldots, i_s$ in the following way. Repeat $t = \mathcal{O}\Big(\|A\|_F^2\|B\|_F^2/\epsilon^2\Big)$ times the following: with probability $\frac{1}{2}$ sample a row index of A and with probability $\frac{1}{2}$ sample a column index of B, according to the usual length-square distribution. Let i denote the sampled index. Accept the index with probability[v] $\frac{2\|A(\cdot,i)\|\|B(i,\cdot)\|}{\|A(\cdot,i)\|^2+\|B(i,\cdot)\|^2}$.

Use the $\leq t$ accepted indices to form a matrix C with the j-th column being $\frac{A(\cdot,i_j)\|A\|_F}{\|A(\cdot,i_j)\|\sqrt{t}}$, and similarly a matrix R with the j-th row being $\frac{B(i_j,\cdot)\|B\|_F}{\|B(\cdot,i_j)\|\sqrt{t}}$. It is easy to see that $\mathbb{E}[tC(\cdot,j)R(j,\cdot)] = AB$, moreover $\mathrm{Var}[tC(\cdot,j)R(j,\cdot)] \leq \mathbb{E}[\|tC(\cdot,j)R(j,\cdot)\|_F^2] \leq \|A\|_F^2\|B\|_F^2$, $\|C\|_F \leq \|A\|_F$, and $\|R\|_F \leq \|B\|_F$. Thus with 0.99 probability we get that $\|AB - CR\|_F \leq \frac{\epsilon}{4}$, see also Lemma B.2.

Compute a matrix Q using Proposition 3.4, which estimates $C^\dagger C$ such that $\big\|Q - C^\dagger C\big\| \leq \xi := \delta\frac{\epsilon^2}{32\|B\|_F^2}$ with 0.99 probability. Similarly compute a matrix T such that $\big\|T - RR^\dagger\big\| \leq \xi$ with 0.99 probability. Let $\bar{U}$ be the matrix whose columns are formed by those (orthonormalized) eigenvectors of Q that have eigenvalues less than $\frac{\epsilon^2}{32\|B\|_F^2}$. Observe that

$$\big\|C\bar{U}\big\|^2 = \Big\|\bar{U}^\dagger C^\dagger C\bar{U}\Big\| \leq \Big\|\bar{U}^\dagger(C^\dagger C - Q)\bar{U}\Big\| + \Big\|\bar{U}^\dagger Q\bar{U}\Big\| \leq \xi + \frac{\epsilon^2}{32\|B\|_F^2} \leq \frac{\epsilon^2}{16\|B\|_F^2}. \tag{46}$$

Let $\hat{U}$ be the matrix whose columns are formed by those (orthonormalized) eigenvectors of Q that have eigenvalues at least $\frac{\epsilon^2}{32\|B\|_F^2}$, and define $\bar{V}, \hat{V}$ analogously. Then we have that

$$\begin{aligned}
\Big\|CR - C\hat{U}\hat{U}^\dagger\hat{V}\hat{V}^\dagger R\Big\|_F &= \Big\|C(\hat{U}\hat{U}^\dagger + \bar{U}\bar{U}^\dagger)(\hat{V}\hat{V}^\dagger + \bar{V}\bar{V}^\dagger)R - C\hat{U}\hat{U}^\dagger\hat{V}\hat{V}^\dagger R\Big\|_F &\quad (47)\\
&\leq \Big\|C\bar{U}\bar{U}^\dagger R\Big\|_F + \Big\|C\hat{U}\hat{U}^\dagger\bar{V}\bar{V}^\dagger R\Big\|_F &\quad (48)\\
&\leq \big\|C\bar{U}\big\|\|R\|_F + \|C\|_F\Big\|\bar{V}^\dagger R\Big\| \leq \frac{\epsilon}{2}. &\quad (49)
\end{aligned}$$

This implies that $\Big\|AB - C\hat{U}\hat{U}^\dagger\hat{V}\hat{V}^\dagger R\Big\|_F \leq \frac{3}{4}\epsilon$.

Let D_Q be a diagonal matrix containing the eigenvalues of Q corresponding to the eigenvectors in $\hat{U}$, and similarly define D_T. Then $D_Q^{-\frac{1}{2}}\hat{U}^\dagger Q\hat{U}D_Q^{-\frac{1}{2}} = I$, and thus $\Big\|D_Q^{-\frac{1}{2}}\hat{U}^\dagger C^\dagger C\hat{U}D_Q^{-\frac{1}{2}} - I\Big\| \leq$

[iv] Otherwise simply rescale the matrices by $A \leftarrow \sqrt{\frac{\|B\|_F}{\|A\|_F}}$ and $B \leftarrow \sqrt{\frac{\|A\|_F}{\|B\|_F}}$.

[v] Note that this number is always in $[0,1]$, as shown by the inequality of arithmetic and geometric means.

$\xi\frac{32\|B\|_F^2}{\epsilon^2} = \delta$. Now we have

$$\left\| AB - \left(C\hat{U}D_Q^{-\frac{1}{2}}\right)\left(D_Q^{\frac{1}{2}}\hat{U}^\dagger\hat{V}D_T^{\frac{1}{2}}\right)\left(D_T^{-\frac{1}{2}}\hat{V}^\dagger R\right)\right\|_F \le \frac{3}{4}\epsilon. \tag{50}$$

Finally, compute the singular value decomposition of $D_Q^{\frac{1}{2}}\hat{U}^\dagger\hat{V}D_T^{\frac{1}{2}} = \tilde{U}D\tilde{V}$, and define $U := C\hat{U}D_Q^{-\frac{1}{2}}\tilde{U} = \left(\frac{\sqrt{t}}{\|A\|_F}C\right)\left(\frac{\|A\|_F}{\sqrt{t}}\hat{U}D_Q^{-\frac{1}{2}}\tilde{U}\right)$ and $V := \tilde{V}D_T^{-\frac{1}{2}}\hat{V}^\dagger R = \left(\frac{\|B\|_F}{\sqrt{t}}\tilde{V}D_T^{-\frac{1}{2}}\hat{V}^\dagger\right)\left(\frac{\sqrt{t}}{\|B\|_F}R\right)$ as the output matrices.

For the complexity analysis note that the initial index sampling has cost $\mathcal{O}\big(t(\mathbf{sq}(A^\dagger) + \mathbf{sq}(B))\big)$. Then, in Proposition 3.4 we should use precision $\frac{\delta}{t}\frac{\epsilon^2}{32\|B\|_F^2}$, which leads to complexity

$$\mathcal{O}\left(\frac{s^2}{\delta^2}(\mathbf{sq}(A^\dagger) + \mathbf{sq}(B))\right). \tag{51}$$

Finally, the cost of computing the spectral decomposition and the singular-value decomposition is upper bounded by $\mathcal{O}(t^3)$. □

4 Applying the framework to dequantizing QML algorithms

Now, with our framework, we can recover previous dequantization results: recommendation systems (Section 4.2), principal component analysis (Section 4.3), supervised clustering (Section 4.4), low-rank matrix inversion (Section 4.5), low-rank semidefinite programs (Section 4.7), and support-vector machines (Section 4.8). We also propose new quantum-inspired algorithm for other applications, including Hamiltonian simulation (Section 4.1) and discriminant analysis (Section 4.6).

4.1 Hamiltonian simulation

The problem of simulating the dynamics of quantum systems was the original motivation for quantum computers proposed by Feynman [Fey82], and it has wide applications in quantum physics, quantum chemistry, etc. Specifically, given a Hamiltonian H a quantum state $|\psi\rangle$, and a time $t > 0$, it asks to prepare a quantum state $|\psi_t\rangle$ such that

$$\||\psi_t\rangle - e^{iHt}|\psi\rangle\| \le \epsilon \tag{52}$$

where $\epsilon \in (0,1)$ is an error parameter. Since then, there has been rich literature on Hamiltonian simulation [Llo96, ATS03, BCC+14, BCC+15, BCK15], with an optimal quantum algorithm for simulating sparse Hamiltonians given in [LC17].

In this subsection, we consider classical algorithms for Hamiltonian simulation using our matrix arithmetic tools. Specifically, we ask:

Problem 4.1. For a Hermitian matrix $H \in \mathbb{R}^{n\times n}$, a unit vector $b \in \mathbb{R}^n$, and a time $t > 0$, given $\mathrm{SQ}(H)$ and $\mathrm{SQ}(b)$, (approximately) respond to queries for $\mathrm{SQ}(v)$ where $\|v - e^{itH}b\| \le \epsilon$.

Corollary 4.2. *Suppose that $H \in \mathbb{C}^{n\times n}$ is a Hermitian matrix. There is an algorithm solving Problem 4.1 in $\widetilde{\mathcal{O}}\left(\frac{t^{36}\|H\|_F^{36}}{\epsilon^{36}}\right)$ time.*

Proof. Let $f(x) = e^{itx}$. Since $|f'(x)| = |ite^{itx}| = t$ we have that f is t-Lipschitz. By taking $\alpha = t$, $\epsilon' = \epsilon/2$, and f in Theorem 3.3, we get an approximate version B of e^{itH} such that $\|(I+B) - e^{itH}\| \le \epsilon/2$, where $B = \check{U}D\check{V}^\dagger$ for $\check{U}, \check{V} \in \mathbb{C}^{n\times r}$ with $r = \widetilde{\mathcal{O}}\left(\frac{t^2\|H\|_F^2}{\epsilon^2}\right)$, and $\mathrm{SQ}(\check{U}), \mathrm{SQ}(\check{V})$ in time complexity $\widetilde{\mathcal{O}}\left(\frac{t^{36}\|H\|_F^{36}}{\epsilon^{36}}\right)$. We next apply Proposition 3.7, which obtains $\mathrm{SQ}(b_t)$ such that $\|b_t - (I+B)b\| \le \epsilon/2$, where the time complexity is dominated by that of Theorem 3.3. Note that following the analysis of [Tan19] and the choice of σ in the proof of Theorem 3.3, the value of C_{S,u_j} and C_{T,v_j} in Proposition 3.7 are bounded by $\widetilde{\mathcal{O}}\left(\|H\|_F^2 t^2/\epsilon^2\right)$. In all,

$$\|b_t - e^{itH}b\| \le \|b_t - (I+B)b\| + \|(I+B) - e^{itH}\| \le \epsilon, \tag{53}$$

and the total time complexity is $\widetilde{\mathcal{O}}\left(\frac{t^{36}\|H\|_F^{36}}{\epsilon^{36}}\right)$. ☐

Remark 4.3. Note that Ref. [RWC+18] essentially achieves the same result up to polynomial factors. Compared to their result, our algorithm has the feature that the time and the Hamiltonian norm have the same exponent; moreover, for the vector $b \in \mathbb{R}^n$ we only assume $\mathrm{SQ}(b)$, whereas their result requires sparsity of b in addition.

4.2 Recommendation systems

Tang's dequantization [Tan19] of Kerenidis and Prakash's recommendation system [KP17] is the first dequantization in this line of work that uses techniques from randomized linear algebra.

We want to find a product $j \in [n]$ that is a good recommendation for a particular user $i \in [m]$, given incomplete data on user-product preferences. If we store this data in a matrix $A \in \mathbb{R}^{m\times n}$ with sampling and query access, in the strong model described by Kerenidis and Prakash [KP17], finding good recommendations reduces to the following:

Problem 4.4. For a matrix $A \in \mathbb{R}^{m\times n}$, given $\mathrm{SQ}(A)$ and a row index $i \in [m]$, sample from $\tilde{A}(i,\cdot)$ up to ϵ error in total variation distance, where $\|\tilde{A} - A_{\sigma,1/6}\|_F \le \epsilon\|A\|_F$.

Corollary 4.5. *A classical algorithm can solve Problem 4.4 in* $\widetilde{\mathcal{O}}\left(\frac{\|A\|_F^{24}}{\epsilon^{12}\sigma^{24}}\right)$ *time.*

Proof. Note that $A_{\sigma,1/6}$ is $f(A)$, where f is the singular value transformation

$$f(x) = \begin{cases} 0 & x < \frac{5\sigma}{6} \\ \frac{7}{2}(x - \frac{5}{6}\sigma) & \frac{5}{6}\sigma \le x < \frac{7}{6}\sigma \\ x & \frac{7}{6}\sigma \le x. \end{cases} \tag{54}$$

Note that f is 7/2-Lipschitz continuous, and Theorem 3.1 can be used to solve this problem. However, a simpler proof is possible by directly using Theorem 5.1, which yields the desired $\tilde{A}$ in the form $\mathrm{SQ}(U^T)\ D$, and $\mathrm{SQ}(V^T)$ in time complexity $\widetilde{\mathcal{O}}\left(\frac{\|A\|_F^{24}}{\epsilon^{12}\sigma^{24}}\right)$. Then we use Proposition 3.7 to sample from $\tilde{A}(i,\cdot) = U(i,\cdot)DV^T$. The time complexity is dominated by the cost of Theorem 5.1. Note that following the analysis of [Tan19] and the choice of σ in the proof of Theorem 3.3, the value of C_{S,u_j} and C_{T,v_j} in Proposition 3.7 are bounded by $\widetilde{\mathcal{O}}\left(\|A\|_F^2/\sigma^2\right)$. ☐

4.3 Principal component analysis

Principal component analysis (PCA) is an important data analysis tool, first proposed to be feasible via quantum computation by Lloyd et al. [LMR14]. The standard result is that of *spectral sampling*, which can be stated as the following problem:

Problem 4.6 (Quantum spectral sampling)**.** Given a matrix $X \in \mathbb{R}^{m\times n}$ such that $X^T X$ has eigenvalues $\{\lambda_i\}_{i=1}^n$ and eigenvectors $\{v_i\}_{i=1}^n$, output λ_i up to additive error $\epsilon \operatorname{Tr}(X^T X)$ and $|v_i\rangle$ with probability $\lambda_i / \operatorname{Tr}(X^T X)$.

The quantum algorithm in [LMR14] does this in $\widetilde{\mathcal{O}}\big(1/\epsilon^3\big)$ cost given access to copies of the density matrix $X^T X / \operatorname{Tr}(X^T X)$; see also Prakash's PhD thesis [Pra14, Section 3.2] for a full analysis, and a variant with boosted success probability. In this subsection, we do not dequantize this protocol exactly, since naive approaches require fine-grained distinguishing between eigenvalues. However, the only useful poly-logarithmic time application that Lloyd et al. [LMR14] suggested for quantum PCA is under the low-rank regime.

Problem 4.7 (PCA for low-rank matrices)**.** Given a matrix $X \in \mathbb{R}^{m\times n}$ with its sample and query access $\mathrm{SQ}(X)$ such that $\operatorname{rank}(X^T X) = k$ with eigenvalues $\{\lambda_i\}_{i=1}^k$ and eigenvectors $\{v_i\}_{i=1}^k$, compute $\lambda_1, \dots, \lambda_k$ and $\mathrm{SQ}(v_1, \dots, v_k)$ up to $\epsilon \operatorname{Tr}(X^T X)$ error.

Without loss of generality, assume $\lambda_1 \geq \cdots \geq \lambda_k > 0$. The reason this setting is interesting is that, when $\lambda_k = \Omega(\operatorname{Tr}(X^T X))$, we can actually learn the spectrum of $X^T X$ from the PCA estimate. Note that, to robustly avoid degeneracy conditions, our runtime must depend on

$$K := \operatorname{Tr}(X^T X)/\lambda_k \geq k \quad \text{and} \quad \eta := \min_{i\in[k]} |\lambda_i - \lambda_{i+1}| / \operatorname{Tr}(X^T X); \tag{55}$$

η can be regarded as the spectral gap. This low-rank setting reduces to spectral sampling:

Lemma 4.8. *If we can solve spectral sampling (Problem 4.6) in $f(1/\epsilon)$ time, we can solve Problem 4.7 in $\widetilde{\mathcal{O}}(f(1/\epsilon\eta)\cdot K)$ time.*

Proof. For the spectral sampling, we set ϵ to be $\Theta(\epsilon\eta)$, so we can always distinguish whether two estimates α, β come from the same λ_i's or different ones with high success probability using Chernoff's bound. Since for all $i \in [k]$, the probability of each λ_i shows up is at least $1/K$, we can learn all of the λ_i's as a coupon collector problem with $O(K \log K)$ samples. Furthermore, if a request for $\mathrm{SQ}(v_i)$ is desired, we run spectral sampling until we see an estimate for λ_i. □

This regime is the only one where spectral sampling allows us to learn about a particular eigenvector or eigenvalue. If $X^T X$ is far from low-rank, say $\lambda_i = \operatorname{Tr}(X^T X)/\operatorname{poly}(n)$ for all i, then distinguishing a λ_i from λ_{i+1} necessarily takes $\operatorname{poly}(n)$ samples, and even sampling the same i twice takes $\operatorname{poly}(n)$ samples, so learning v_i is also impossible. We can solve this low-rank PCA problem easily, as first noted in [Tan18].

Corollary 4.9. *We can solve Problem 4.7 in $\widetilde{\mathcal{O}}\Big(\frac{K}{(\epsilon\eta)^{18}}\Big)$ time.*

Proof. By Lemma 4.8, it remains to solve spectral sampling (Problem 4.6) in $\widetilde{\mathcal{O}}\big(\frac{1}{\epsilon^{18}}\big)$ time.

Without loss of generality, assume $\operatorname{Tr}(X^T X) = 1$. Denote the singular value decomposition of X as $X = U\Sigma V$; then $X^T X = (U\Sigma V)^T U\Sigma V = V^T \Sigma^2 V$. As a result, we can apply Theorem 3.1 with the function $f(x) = x^2$ and $\alpha = 2$ (because f in $[0,1]$ is 2-Lipschitz), $\epsilon' = \sqrt{\epsilon}$; also note that $\|X\|_F = \operatorname{Tr}(X^T X) = 1$. Taking a sample from $\mathrm{SQ}(\Sigma)$ and the corresponding column of V exactly gives the spectrum sampling as needed in Problem 4.6. □

4.4 Supervised clustering

The 2013 paper of Lloyd et al. [LMR13] gives two algorithms for the machine learning problem of clustering. The first algorithm is a simple swap test procedure that was dequantized by Tang [Tan18] (the second is an application of the quantum adiabatic algorithm with no proven runtime guarantees). Since the dequantization is very simple, only using the inner product protocol, it rather trivially fits into our framework.

We have a dataset of points in $\mathbb{R}^n$ grouped into clusters, and we wish to classify a new data point by assigning it to the cluster with the nearest average, aka *centroid*. We do this by estimating the distance between the new point $p \in \mathbb{R}^n$ to the centroid of a cluster of points $q_1, \ldots, q_{m-1} \in \mathbb{R}^n$, which reduces to compute $\|Mw\|$ for

$$M := \left[\frac{p}{\|p\|}, \frac{-q_1}{\|q_1\|}, \ldots, \frac{-q_{m-1}}{\|q_{m-1}\|}\right] \in \mathbb{R}^{n\times m}, \qquad w := \left[\|p\|, \frac{\|q_1\|}{m-1}, \ldots, \frac{\|q_{m-1}\|}{m-1}\right]^T \in \mathbb{R}^m. \tag{56}$$

If we assume sample and query access to the data points, computing $\|p - \frac{1}{m-1}(q_1 + \cdots + q_{m-1})\|^2$ reduces to [Tan18]:

Problem 4.10. For $M \in \mathbb{R}^{n\times m}, w \in \mathbb{R}^m$, and $\mathrm{SQ}(M^T, w)$, approximate $(Mw)^T(Mw)$ to additive ϵ error with probability at least $1-\delta$.

Corollary 4.11. *There is a classical algorithm that can solve Problem 4.10 in $\mathcal{O}\big(\|M\|_F^2\|w\|\frac{1}{\epsilon^2}\log\frac{1}{\delta}\big)$ time.*

Proof. Let $\overline{M} := [\|M(\cdot,1)\|, \ldots, \|M(\cdot,n)\|]$. We can rewrite $(Mw)^T(Mw)$ as an inner product $\langle u, v\rangle$ where

$$u := \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{m} M(i,j)\|M(\cdot,k)\| e_i \otimes e_j \otimes e_k = M \otimes \overline{M} \tag{57}$$

$$v := \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{m} \frac{w_j w_k M(i,k)}{\|M(\cdot,k)\|} e_i \otimes e_j \otimes e_k, \tag{58}$$

where u and v are three-dimension tensors. By flattening u and v, we can represent them as two vectors in $\mathbb{R}^{(m\cdot n\cdot n)\times 1}$. In the following, we show that we can get $\mathrm{SQ}(u)$ and $\mathrm{SQ}(v)$ in constant time.

It is easy to get $\mathrm{SQ}(u)$ from $\mathrm{SQ}(M^T)$: for the sampling access, we first sample i according to $\|M(\cdot,i)\|^2/\|M\|_F^2$, sample j according to $\|M(j,i)\|^2/\|M(\cdot,i)\|^2$, and then sample k according to $\overline{M}(k)^2/\|\overline{M}\|^2$; we can get $u_{i,j,k}$ be computing $M(i,j)\overline{M}(k)$. The query access to w and M directly give the query access to v. Finally, we can apply Proposition 3.4 to estimate $\langle u, v\rangle$. $\|u\| = \|M\|_F^2$ and $\|v\| = \|w\|^2$, so estimating $\langle u, v\rangle$ to ϵ additive error with probability at least $1-\delta$ requires $O(\frac{\|M\|_F^2\|w\|}{\epsilon})\log\frac{1}{\delta}$. $\square$

4.5 Matrix inversion and principal component regression

The low-rank matrix inversion algorithm given by Gilyén et al. and Chia et al. [GLT18, CLW18] dequantizes Harrow, Hassidim, and Lloyd's quantum matrix inversion algorithm (HHL) [HHL09] in the regime where the input matrix is *low-rank* instead of sparse – the corresponding quantum algorithm is present in the work of Rebentrost et al. [RSML18]. Since sparse matrix inversion is

BQP-complete, it is unlikely that one can efficiently dequantize it. Nevertheless, low-rank (non-sparse) matrix inversion is at the core of many quantum machine learning papers, making it a very influential primitive. Using the presented framework it is straightforward to derive the low-rank matrix inversion algorithm similar to those of [GLT18, CLW18]. Moreover, we can also handle the approximately low-rank regime and only invert the matrix on a well-conditioned subspace, solving principle component regression – for more discussion see [GSLW19]. Let us define the threshold pseudoinverse $A^+_{\theta,\xi}$ as a matrix that equals A^+ on the subspace of (left) singular vectors belonging to singular values that are at least θ, it has $\left\|A^+_{\theta,\xi}\right\| = \frac{1}{\theta}$, and its kernel contains the (left) singular vectors corresponding to singular values that are at most $\theta(1-\xi)$. Note that if $\theta \leq \frac{1}{\|A^+\|}$, then $A^+_{\theta,\xi} = A^+$.

Theorem 4.12 (Implementing the threshold pseudoinverse)**.** *Let $A = A^{(1)} + \cdots + A^{(\tau)}$ where for each $A^{(\ell)} \in \mathbb{C}^{m\times n}$, we have $\left\|A^{(\ell)}\right\| \leq 1$. For $\epsilon', \xi, \theta \in (0,1)$, $\eta = \widetilde{\mathcal{O}}\big(\epsilon'^2\xi^2\theta^4/(\sum_\ell \|A^{(\ell)}\|_F^2)\big)$ there exists an algorithm with time complexity $\widetilde{\mathcal{O}}\left(\left(\frac{\tau(\sum_\ell \|A^{(\ell)}\|_F^2)}{\epsilon'^2\xi^2\theta^4}\right)^{18} \mathbf{sq}(A^{[\tau]^{(\dagger)}})\right)$ which provides succinct descriptions of $\widetilde{\mathcal{O}}\left(\frac{\epsilon'^2\xi^2\theta^4}{\|A\|_F^2}\right)$-approximate isometries $\check{V}$, $\check{U}$, and a diagonal matrix $D \in \mathbb{R}^{r\times r}$ such that the matrix $B := \check{U}D\check{V}^\dagger$ satisfies $\left\|B - A^+_{\theta,\xi}\right\| \leq \epsilon'$ with probability $9/10$. Moreover, the provided representation of B enables $\mathrm{SQ}^\eta_\eta(B)$ access in time $\mathbf{s}(B) = \mathcal{O}\big(r^2\mathbf{sq}(\check{U},\check{V})\big)$, $\mathbf{q}(B) = \mathcal{O}\big(r\mathbf{sq}(\check{U},\check{V})\big)$, $\mathbf{n}_\eta(B) = \mathbf{n}^\eta(B) = \mathcal{O}\big(r^2/\eta^2\mathbf{sq}(\check{U},\check{V})\big)$, where $r \leq \ell_{\frac{A}{\|A\|_F}}(\widetilde{\Theta}(\frac{\epsilon'\xi\theta^2}{\|A\|_F})) = \widetilde{\mathcal{O}}\left(\frac{\|A\|_F^2}{\epsilon'^2\xi^2\theta^4}\right)$ and $\mathbf{sq}(\check{U},\check{V}) = \widetilde{\mathcal{O}}\left(\left(\frac{\tau(\sum_\ell \|A^{(\ell)}\|_F^2)}{\epsilon'^2\xi^2\theta^4}\right)^{13} \mathbf{sq}(A^{[\tau]^{(\dagger)}})\right)$.*

Moreover, in case A is Hermitian, we can also have input in the form $A = \gamma I + A^{(1)} + \cdots + A^{(\tau)}$. In this case the output B is such that $\left\|\mathrm{sign}(\gamma)f(|\gamma|)I + B - A^+_{\theta,\xi}\right\| \leq \epsilon'$.

Proof. Apply Theorem 3.1 to $A^\dagger$ with the $\frac{1}{\theta\xi}$-Lipschitz function

$$f(x) = \begin{cases} 0, & \text{for} \quad 0 \leq x \leq \theta(1-\xi) \\ \frac{1}{\xi\theta^2}(x - \theta(1-\xi)), & \text{for } \theta(1-\xi) \leq x \leq \theta \\ \frac{1}{x}, & \text{for} \quad \theta \leq x. \end{cases} \tag{59}$$

Then the complexity claim directly follows. ☐

Using the above succinct representation of the (threshold) pseudo-inverse, we can solve regression problems and perform various other important tasks.

4.6 Discriminant analysis

Discriminant analysis is used for dimensionality reduction and classification over large data sets. A qunatum algoritm for this problem was introduced in [CD16]. We roughly follow their work in this subsection.

Discriminant analysis is similar in spirit to the widely used principal components analysis (PCA) technique, which finds the directions in the data of maximal variance. A drawback of PCA is that it only considers the global data variance, without taking inot account the class data. Fisher's linear discriminant analysis (LDA) is a refined approach which aims to overcome this problem by maximizing the between-class variance, while minimizing the variance within the classes. As a result LDA can be more effective than PCA for dimensionality reduction and classification.

Suppose there are M input data points $\{x_i \in \mathbb{R}^N : 1 \le i \le M\}$ each belonging to one of k classes. Let μ_c denote the centroid (mean) of class $c \in [k]$, and $\bar{x}$ denote the centroid of all data points. Following the notation of [CD16], let

$$S_B = \sum_{c=1}^{k} (\mu_c - \bar{x})(\mu_c - \bar{x})^T \tag{60}$$

denote the between-class scatter matrix of the dataset, and let

$$S_W = \sum_{c=1}^{k} \sum_{x \in c} (\mu_c - x)(\mu_c - x)^T. \tag{61}$$

Using this notation, the LDA dimensionality reduction procedure can be described as follows:

Problem 4.13 (Dimensionality reduction)**.** Find the largest p eigenvalues and eigenvectors of $S_W^{-1} S_B$.

One usually first uses dimensionality reduction, and then uses some standard classification method, but it is also possible to perform classification directly using the so-called discriminant functions. For brevity we only discuss dimensionality reduction, but direct classification can be performed in a similar manner, for more details we refer to [CD16].

Now we turn to solving the above problem. Note that the matrix $S_W^{-1} S_B$ is similar to $S_B^{\frac{1}{2}} S_W^{-1} S_B^{\frac{1}{2}}$, i.e., if $S_B^{\frac{1}{2}} S_W^{-1} S_B^{\frac{1}{2}} v = \lambda v$, then $S_W^{-1} S_B (S_B^{-\frac{1}{2}} v) = \lambda (S_B^{-\frac{1}{2}} v)$, so it suffices to find the eigenvectors of $S_B^{\frac{1}{2}} S_W^{-1} S_B^{\frac{1}{2}}$. Moreover the eigenvectors of $S_B^{\frac{1}{2}} S_W^{-1} S_B^{\frac{1}{2}}$ coincide with the right singular vectors of $S_W^{-\frac{1}{2}} S_B^{\frac{1}{2}}$.

It might be the case that S_B and S_W have very large condition numbers, therefore we use a thresholded version of the (matrix) function $\frac{1}{\sqrt{x}}$ with threshold θ, like in Theorem 4.12 (say choosing $\eta = 1/2$). Also the function $\sqrt{x}$ is not Lipschitz, but we can ϵ-approximate it with an $\mathcal{O}(1/\epsilon)$-Lipschitz function. So we apply Theorem 3.3 to get a succinct description of approximate (thresholded) versions of $S_W^{-\frac{1}{2}}$ and $S_B^{\frac{1}{2}}$. Via Theorem 3.9 this also yields a succinct description of $S_W^{-\frac{1}{2}} S_B^{\frac{1}{2}}$, which can be used for recovering the (right) singular vectors. Finally we can apply the map $S_B^{-\frac{1}{2}}$ to these vectors to get the final output. Given $SQ(X_c)$ access to the clustered data points[vi], all of these steps can be efficiently performed using our sample and query framework, solving the problem up to ϵ precision in complexity $\mathcal{O}\big(\mathrm{poly}(\|X\|_F, \epsilon^{-1}, \theta^{-1}) \mathbf{sq}(X_{[k]})\big)$.

4.7 Low-rank semidefinite programs

Semidefinite program (SDP) is a central topic in the research of mathematical optimization, with a wide range of applications including algorithm design, operations research, machine learning, etc. Specifically, an SDP problem is defined as follows:

[vi] Note that the centroids can be obtained via a simple matrix vector product, which can be also efficiently computed in our framework.

Problem 4.14 (Semidefinite program)**.** Given $m \in \mathbb{N}$, $b_1, \ldots, b_m \in \mathbb{R}$, and $n \times n$ Hermitian matrices $A^{(1)}, \ldots, A^{(m)}, C$ where $-I \preceq A^{(i)}, C \preceq I\ \forall i \in [m]$, denote OPT to be

$$\begin{aligned} \max \quad & \operatorname{Tr}[CX] && (62)\\ \text{s.t.} \quad & \operatorname{Tr}[A^{(i)}X] \leq b_i \quad \forall\, i \in [m]; && (63)\\ & X \succeq 0, && (64) \end{aligned}$$

The problem is to output an X^* such that Eqs. (63) and (64) are satisfied while $\operatorname{Tr}[CX^*] \geq \mathsf{OPT} - \varepsilon$.

It is a well-known fact that one can use binary search to reduce ϵ-approximation of the SDP in Eqs. (62) to (64) to $O(\log(1/\epsilon))$ calls of the following feasibility problem[vii]:

Definition 4.15 (Feasibility of SDP)**.** Given an $\epsilon > 0$, m real numbers $a_1, \ldots, a_m \in \mathbb{R}$, and Hermitian $n \times n$ matrices $A^{(1)}, \ldots, A^{(m)}$ where $-I \preceq A^{(i)} \preceq I, \forall j \in [m]$, define $\mathcal{S}_\epsilon$ as the set of all X such that

$$\begin{aligned} \operatorname{Tr}[A^{(i)}X] \leq a_i + \epsilon \quad \forall\, i \in [m]; && (65)\\ X \succeq 0; && (66)\\ \operatorname{Tr}[X] = 1. && (67) \end{aligned}$$

For ϵ-approximate feasibility testing of the SDP, we require that:

- If $\mathcal{S}_\epsilon = \varnothing$, output "infeasible";
- If $\mathcal{S}_0 \neq \varnothing$, output an $X \in \mathcal{S}_\epsilon$.[viii]

Throughout the subsection, we focus on solving feasibility testing of SDPs. This relies on the matrix multiplicative weight (MMW) framework (see e.g. [AK07, Kal07, LRS15, BKL+19, CLLW19]). To be more specific, MMW works as a zero-sum game with two players, where the first player wants to provide an $X \in \mathcal{S}_\epsilon$, and the second player wants to find any violation $j \in [m]$ of any proposed X, i.e., $\operatorname{Tr}[A^{(j)}X] > a_j + \epsilon$. At the t^{th} round of the game, if the second player points out a violation j_t for the current solution X_t, the first player proposes a new solution

$$X_{t+1} \leftarrow \exp[-\epsilon(A^{(j_1)} + \cdots + A^{(j_t)})], \tag{68}$$

up to normalization. Such solution is formally known as a *Gibbs state*. It is proved in [BKL+19] that $T = \frac{16 \ln n}{\epsilon^2}$ iterations suffice to solve the SDP feasibility problem with high success probability.

In this subsection, we solve low-rank SDPs by sampling using the MMW framework:

Corollary 4.16. *Given Hermitian matrices $\{A^{(1)}, \ldots, A^{(m)}\}$ with the promise that each of $A^{(1)}, \ldots, A^{(m)}$ has rank at most r, spectral norm at most 1, and the sample and query access*

[vii] We first guess a candidate value $c_1 = 0$ for the objective function, and add that as a constraint $\operatorname{Tr}[CX] \geq c_1$ to the SDP. If this SDP is feasible, the optimum is larger than c_1 and we accordingly take $c_2 = c_1 + \frac{1}{2}$; if this SDP is infeasible, the optimum is smaller than c_1 and we accordingly take $c_2 = c_1 - \frac{1}{2}$; we proceed similarly for all c_i. As a result, we could solve the SDP with precision ϵ using $\lceil \log_2 \frac{1}{\epsilon} \rceil$ calls to the feasibility problem in Definition 4.15.

[viii] If $\mathcal{S}_\epsilon \neq \varnothing$ and $\mathcal{S}_0 = \varnothing$, either output is acceptable.

of each $A^{(i)}$ is given as in Definition 2.6. Also given $a_1, \ldots, a_m \in \mathbb{R}$. Then for any $\epsilon > 0$, Algorithm 1 constructs the sample and query access of the solution of the SDP feasibility problem

$$\operatorname{Tr}[A^{(i)}X] \leq a_i + \epsilon \quad \forall\, i \in [m]; \tag{69}$$

$$X \succeq 0; \tag{70}$$

$$\operatorname{Tr}[X] = 1 \tag{71}$$

with probability at least 2/3 in $\mathcal{O}(\mathrm{poly}(m, \tau, 1/\epsilon, 1/\sigma, 1/\eta))$ time.

Algorithm 1: Feasibility testing of SDPs by our sample-based approach.

---

1 Set the initial Gibbs state $\rho_1 = \frac{I_n}{n}$, and number of iterations $T = \frac{16 \ln n}{\epsilon^2}$;

2 **for** $t = 1, \ldots, T$ **do**

3 Find a $j_t \in [m]$ such that $\operatorname{Tr}[A^{(j_t)}\rho_t] > a_{j_t} + \epsilon$ using Remark 3.5. If we cannot find such j_t, claim that $\rho_t \in \mathcal{S}_\epsilon$ and return $\mathrm{SQ}(\rho_t)$;

4 Otherwise, define the new weight matrix $W_{t+1} := \exp[-\frac{\epsilon}{4}\sum_{i=1}^{t} A^{(j_i)}]$ and Gibbs state $\rho_{t+1} := \frac{W_{t+1}}{\operatorname{Tr}[W_{t+1}]}$;

5 **end**

6 Claim that the SDP is infeasible and terminate the algorithm;

Proof. The correctness of Algorithm 1 directly follows from Theorem 3 of [BKL+19]. It remains to establish the time complexity claim.

The Gibbs state in step 4 is a constant-Lipschitz continuous function on W_i. Therefore, given $\mathrm{SQ}(A^{(1)}), \ldots, \mathrm{SQ}(A^{(\tau)})$, we first estimate $\operatorname{Tr}[W_{i+1}]$ as in [CLLW19] and obtain the query and sampling access to the W_{t+1} in step 4 in the required time according to Theorem 5.1. Then, with $\mathrm{SQ}(A^{(j_t)})$ and query access to the Gibbs state ρ_{i+1}, we can compute $\operatorname{Tr}[A^{(j_t)}\rho]$ with error $\epsilon\sqrt{\frac{m}{\eta}}$ in the required time by Remark 3.5. Finally, we rescale the error from $\epsilon\sqrt{\frac{m}{\eta}}$ to ϵ and repeat the process $T = \tilde{\mathcal{O}}(\frac{1}{\epsilon^2})$ times. □

4.8 Support vector machines

Support vector machine (SVM) is an important technique for classification with wide applications in supervised learning. A quantum algorithm for solving SVM was first introduced in [RML14]. In this paper, we present a dequantization of this quantum algorithm[ix].

Mathematically, the support vector machine is a simple machine learning model attempting to label points in $\mathbb{R}^m$ as $+1$ or -1. Given input data points $x_1, \ldots, x_m \in \mathbb{R}^n$ and their corresponding labels $y_1 \ldots, y_m = \pm 1$. Let $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ be the specification of hyperplanes separating these points. It is possible that no such hyperplane satisfies all the constraints. To resolve this, we add a slack vector $e \in \mathbb{R}^m$ such that $e(j) \geq 0$ for $j \in [m]$. We want to minimize the squared norm of the residuals:

[ix] Another dequantization was reported in [DBH19], but the correctness of some of its technical details is unclear. In particular, Ref. [RML14] gives a quantum algorithm for soft-margin SVMs, but the dequantization in Section VI.C of [DBH19] might not give the correct solution as the approximation of the inverse matrix $(X^TX + \frac{1}{\gamma}I)^{-1}$ below Eq. (5) might incur an non-negligible error.

$$\min_{w,b} \quad \frac{\|w\|^2}{2} + \frac{\gamma}{2}\|e\|^2 \tag{72}$$

$$\text{s.t.} \quad y_i(w^T x_i + b) = 1 - e(i), \quad \forall i \in [m]. \tag{73}$$

The dual of this problem is to maximize over the Karush-Kuhn-Tucker multipliers of a Lagrange function, taking partial derivatives of which yields the linear system:

$$\begin{bmatrix} 0 & \vec{1}^T \\ \vec{1} & X^T X + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \tag{74}$$

where, $\vec{1}$ is the all-ones vector and $X = \{x_1, \ldots, x_m\} \in \mathbb{C}^{n \times n}$. Call the matrix in Eq. (74) F. As in [RML14], we assume that the label vector y is normalized, i.e., $y(i) = \frac{1}{\sqrt{m}} y_i$ for $i \in [m]$. In addition, we assume $\|X^T\|_F = \Omega(\sqrt{m})$. We also assume that the minimum nonzero singular value of $X^T X$ is at least $m\epsilon_\kappa$ [x]. We consider solving the linear system in the low-rank setting:

Problem 4.17. Given $\mathrm{SQ}(y)$, $\mathrm{SQ}(X)$, and ϵ_κ, get b, $\mathrm{Q}(\alpha)$ satisfying Eq. (74). We additionally assume that $\gamma \le 1/(m\epsilon_\kappa)$.

Corollary 4.18. *Problem 4.17 can be solved in time* $\widetilde{\mathcal{O}}\Big(\frac{\|X\|_F^{88}}{\epsilon^{26}\epsilon_\kappa^{72}}\Big)$.

Proof. We consider a normalized version of F. Let $\hat{F}$ be defined as $\hat{F} := F/m$. Define $J := \left[\begin{smallmatrix} 0 & \vec{1}^T \\ \vec{1} & 0 \end{smallmatrix}\right]$, $K := X^T X$, $\hat{J} = J/m$, and $\hat{K} = K/m$. We have $\|\hat{J}\|_F = \sqrt{2/m} < 1$, and $\|\hat{K}\|_F = \Omega(1)$, as $\|X\|_F = \Omega(\sqrt{m})$. We further decompose $\hat{F}$ as $\hat{F} = \hat{J} + \hat{K} + \frac{\gamma^{-1}}{m} I - vv^T$, where

$$v = \begin{bmatrix} \gamma^{-1/2}/\sqrt{m} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{75}$$

We first obtain $\mathrm{SQ}(\hat{K})$ from $\mathrm{SQ}(X)$. To achieve this, we use Theorem 3.9, which, with high probability, outputs $\mathrm{SQ}(U), \mathrm{SQ}(V) \in \mathbb{R}^{n \times r}$ and diagonal $D \in \mathbb{R}^{r \times r}$ such that $\|\hat{K} - UDV\|_F \le \epsilon$ for $r = \widetilde{\mathcal{O}}(\|X\|_F^4/\epsilon^2)$. The time complexity of this procedure is $\widetilde{\mathcal{O}}(\|X\|_F^{12}/\epsilon^6)$. We also have $\mathbf{sq}(U) = \mathbf{sq}(V) = \widetilde{\mathcal{O}}(\|X\|_F^8/\epsilon^4)$. Now, use the similar argument as in the proof of Theorem 3.1, we can obtain $\mathrm{SQ}(UDV)$ for $\mathbf{sq}(B) = \widetilde{\mathcal{O}}(r^2 \mathbf{sq}(U)) = \widetilde{\mathcal{O}}(\|X\|_F^{16}/\epsilon^8)$. We use UDV to approximate $\hat{K}$.

To approximate $\hat{F}^{-1}$, we use Theorem 4.12 with $\theta = 2\epsilon_\kappa$ and $\xi = 1/2$, which yields $\mathrm{SQ}(\check{U})$, D in time complexity $\widetilde{\mathcal{O}}\Big(\frac{\|X\|_F^{72}}{\epsilon^{18}\epsilon_\kappa^{72}} \mathbf{sq}(UDV)\Big) = \widetilde{\mathcal{O}}\Big(\frac{\|X\|_F^{88}}{\epsilon^{26}\epsilon_\kappa^{72}}\Big)$. such that $\|m\gamma I + \check{U} D \check{U}^\dagger - \hat{F}^+_{2\epsilon_\kappa, 1/2}\| \le \epsilon$. Note that the $\gamma^{-1} I/m$ term in $\hat{F}$ does not contribute to the time complexity in Theorem 4.12 as $\hat{F}$ is Hermitian. Define $\bar{y} := \left[\begin{smallmatrix} 0 \\ y \end{smallmatrix}\right]$. $\mathrm{Q}(\bar{y})$ can be easily obtained from $\mathrm{Q}(y)$. To find b and α, we consider $\check{U} D \check{U}^\dagger \bar{y}$. Given $\mathrm{SQ}(\check{U})$ and D, we use Proposition 3.7 to obtain $\mathrm{Q}(u)$ for u in the form of $\left[\begin{smallmatrix} \tilde{b} \\ \tilde{\alpha} \end{smallmatrix}\right]$ such that $\|u - \check{U} D \check{U}^\dagger \bar{y}\| \le \epsilon$. This cost is dominated by the time complexity of Theorem 4.12. Note that following the analysis of [Tan19] and the choice of σ in the proof of Theorem 3.1, the value of C_{S,u_j} and C_{T,v_j} in Proposition 3.7 are bounded by $\widetilde{\mathcal{O}}(\|X\|_F^4/(\epsilon^2\epsilon_\kappa^2))$. □

[x] To have a meaningful ϵ_κ, we assume that $\|X^T X\|$ is not too small, because we know that $\|X^T X\|_F^2 = m$ and we want the singular values of $X^T X$ to "concentrate" to a small number of large singular values. This was also assumed in [RML14].

To classify a new point x, we compute $\mathrm{sgn}(x^T X\alpha + b)$, where $\tilde{b}$ and $\mathrm{Q}(\tilde{\alpha})$ suffice by Remark 3.5.

5 Approximating the singular value and spectral decompositions

In this section, we are going to show that given sample and query access to $A^{(1)}, \ldots, A^{(\tau)}$, we can approximately implement the sample and query access to the spectral or singular-value decomposition of $\sum_{i=1}^{\tau} A^{(i)}$. We first show how to get singular-value decomposition.

Theorem 5.1. *Let $A = A^{(1)} + \cdots + A^{(\tau)}$. Given $\mathrm{SQ}(A^{(\ell)})$, $\mathrm{SQ}(A^{(\ell)\dagger})$, a singular value threshold $\sigma > 0$, and error parameters $\epsilon, \eta > 0$, there exists an algorithm which gives D and a succinct description of $\check{U}$ and $\check{V}$ with probability $9/10$ in time complexity $\mathcal{O}\Big(\frac{\tau^{18}(\sum_\ell \|A^{(\ell)}\|_F^2)^{12}}{\epsilon^{12}\sigma^{24}\eta^6}\mathbf{sq}(A^{[\tau]^{(\dagger)}})\Big)$, where $\check{U}$, $\check{V}$, and D satisfy that*

1. *$\check{U} \in \mathbb{C}^{m\times r}, \check{V} \in \mathbb{C}^{n\times r}$ are $\mathcal{O}\big(\eta\epsilon^2/\tau\big)$-approximate isometries, D is a diagonal matrix, $\ell_A(\sigma(1+\eta)) \leq r \leq \ell_A(\sigma(1-\eta)) = \mathcal{O}\Big(\frac{\|A\|_F^2}{\sigma^2(1-\eta)^2}\Big)$, and*
2. *$\|A_{\sigma,\eta} - \check{U}D\check{V}^\dagger\|_F \leq \epsilon\sqrt{\sum_\ell \|A^{(\ell)}\|_F^2}/\sqrt{\eta}$.*

Moreover, we also obtain $\mathrm{SQ}^{\eta\epsilon^2/\tau}_{\eta\epsilon^2/\tau}(\check{V})$ and $\mathrm{SQ}^{\eta\epsilon^2/\tau}_{\eta\epsilon^2/\tau}(\check{U})$, with $\mathbf{q}(\check{V}) = \mathbf{q}(\check{U}) = \mathcal{O}\Big(\frac{\tau^7(\sum_\ell \|A^{(\ell)}\|_F^2)^4}{\epsilon^4\sigma^8\eta^2}\mathbf{q}(A^{[\tau]^{(\dagger)}})\Big)$, $\mathbf{s}(\check{V}) = \mathbf{s}(\check{U}) = \mathcal{O}\Big(\frac{\tau^{13}(\sum_\ell \|A^{(\ell)}\|_F^2)^9}{\epsilon^8\sigma^{18}\eta^4}\mathbf{sq}(A^{[\tau]^{(\dagger)}})\Big)$, $\mathbf{n}^{\eta\epsilon^2/\tau}_{\eta\epsilon^2/\tau}(\check{V}) = \mathbf{n}^{\eta\epsilon^2/\tau}_{\eta\epsilon^2/\tau}(\check{U}) = \mathcal{O}(1)$.

The proof of is postponed to Section 5.2. Theorem 5.1 implies that given $A^{(1)}, \ldots, A^{(\tau)}$, we can approximate the singular-value decomposition of $A := \sum_{i=1}^{\tau} A^{(i)}$ in time sublinear in the dimension of A. By Theorem 5.1, if A is a single matrix, we can also approximate the singular-value decomposition of A.

When $A^{(1)}, \ldots, A^{(\tau)}$ are Hermitian matrices, we can have a simpler algorithm to approximate the spectral decomposition.

Theorem 5.2. *Let $A = A^{(1)} + \cdots + A^{(\tau)}$ where for each $A^{(\ell)} \in \mathbb{C}^{m\times n}$, $A^{(\ell)}$ is Hermitian. Given $\mathrm{SQ}(A^{(\ell)})$, a singular value threshold $\sigma > 0$, and error parameters $\epsilon, \eta > 0$, there exists an algorithm which gives D and a succinct description of $\check{U}$ with probability $9/10$ in time complexity $\mathcal{O}\Big(\frac{\tau^{18}(\sum_\ell \|A^{(\ell)}\|_F^2)^{12}}{\epsilon^{12}\sigma^{24}\eta^6}\mathbf{sq}(A^{[\tau]})\Big)$, where $\check{U}$ and D satisfy that*

1. *$\check{U} \in \mathbb{C}^{n\times r}$ is $\mathcal{O}\big(\eta\epsilon^2/\tau\big)$-approximate isometry, D is a diagonal matrix, $\ell_A(\sigma(1+\eta)) \leq r \leq \ell_A(\sigma(1-\eta)) = \mathcal{O}\Big(\frac{\|A\|_F^2}{\sigma^2(1-\eta)^2}\Big)$, and*
2. *$\|A_{\sigma,\eta} - \check{U}D\check{U}^\dagger\|_F \leq \epsilon\sqrt{\sum_\ell \|A^{(\ell)}\|_F^2}/\sqrt{\eta}$.*

Moreover, we also obtain $\mathrm{SQ}^{\eta\epsilon^2/\tau}_{\eta\epsilon^2/\tau}(\check{U}^\dagger)$, with and $\mathbf{n}^{\eta\epsilon^2/\tau}_{\eta\epsilon^2/\tau}(\check{U}) = \mathcal{O}(1)$ and

$$\mathbf{q}(\check{U}) = \mathcal{O}\left(\frac{\tau^7(\sum_\ell \|A^{(\ell)}\|_F^2)^4}{\epsilon^4\sigma^8\eta^2}\mathbf{q}(A^{[\tau]})\right), \quad \mathbf{s}(\check{U}) = \mathcal{O}\left(\frac{\tau^{13}(\sum_\ell \|A^{(\ell)}\|_F^2)^9}{\epsilon^8\sigma^{18}\eta^4}\mathbf{sq}(A^{[\tau]})\right).$$

The proof of is postponed to Section 5.2. In the following, we give some subroutines which we will use to prove the theorems in this section.

5.1 Sampling a small submatrix

The objective of this section is to provide a method for sampling a small submatrix of A of the form $A = A^{(1)} + \cdots + A^{(\tau)}$ where the sampling access of each $A^{(\ell)}$ is given. A *weighted sampling* method was proposed in [CLLW19] to accommodate this situation, where the intuition is assigning each $A^{(\ell)}$ a different weight when computing the probability distribution, and then sampling a row/column index of A according to this probability distribution. In this paper, we give a simpler proof of the weighted sampling method in [CLLW19].

We first give the method for sampling row indices of A as in Procedure 2. The objective of this procedure is to sample a submatrix S such that $S^\dagger S \approx A^\dagger A$.

Procedure 2: Weighted sampling of rows.

input : $A = \sum_{l=1}^{\tau} A^{(\ell)}$ with $\mathrm{SQ}(A^{(\ell)})$ for $\ell \in [\tau]$; integer p.
output : indices $i_1, \ldots, i_p$; probabilities $P_{i_1}, \ldots, P_{i_p}$

1 Obtain p row indices sampled according to the distribution $P_i = \sum_{j=1}^{\tau} \mathcal{D}_{\mathrm{rows}(A^{(j)})}(i) \|A^{(j)}\|_F^2 / \left(\sum_{\ell=1}^{\tau} \|A^{(\ell)}\|_F^2\right)$ for $i = 1, \ldots, m$ **begin**
2   Sample $j \in [\tau]$ distributed as $\|A^{(j)}\|_F^2 / \left(\sum_{\ell=1}^{\tau} \|A^{(\ell)}\|_F^2\right)$;
3   Sample a row index i according to $\mathcal{D}_{\mathrm{rows}(A^{(j)})}(i)$ using $\mathrm{SQ}(A^{(\ell)})$;
4 **end**
5 For the row indices $i_1, \ldots, i_p$ compute $P_{i_1}, \ldots, P_{i_p}$;

After applying Procedure 2, we obtain the row indices $i_1, \ldots, i_p$. Let $S^{(1)}, \ldots, S^{(\tau)}$ be matrices such that $S^{(\ell)}(t, \cdot) = A^{(\ell)}(i_t, \cdot)/\sqrt{pP_{i_t}}$ for all $t \in [p]$ and $\ell \in [\tau]$. Define the matrix S as

$$S = S^{(1)} + \cdots + S^{(\tau)}. \tag{76}$$

Next, we sample column indices of S as in Procedure 3 in order to sample a submatrix W from S such that $WW^\dagger \approx SS^\dagger$.

Procedure 3: Weighted sampling of columns.

input : $A = \sum_{l=1}^{\tau} A^{(\ell)}$ with $\mathrm{SQ}(A^{(\ell)})$ for $\ell \in [\tau]$; $i_1, \ldots, i_p$ obtained in Procedure 2; integer p.
output : indices $j_1, \ldots, j_p$; probabilities $Q_{1|i_{t_1}}, \ldots, Q_{j_p|i_{t_p}}$

1 Do the following p times independently to obtain samples $j_1, \ldots, j_p$. **begin**
2   Sample a row index $t \in [p]$ uniformly at random ;
3   Sample a column index $j \in [n]$ from the probability distribution $\{Q_{1|i_t}, \ldots, Q_{n|i_t}\}$ where $Q_{j|i_t} = \sum_{k=1}^{\tau} \mathcal{D}_{A^{(k)}(i_t,\cdot)}(j) \left\|A^{(k)}(i_t, \cdot)\right\|^2 / \left(\sum_{\ell=1}^{\tau} \left\|A^{(\ell)}(i_t, \cdot)\right\|^2\right)$;
4 **end**
5 For the column indices as $j_1, \ldots, j_p$, compute $Q_{j_1|i_{t_1}}, \ldots, Q_{j_p|i_{t_p}}$ where $t_1, \ldots, t_p$ are sampled in Line 2;

Once we have obtained column indices $j_1, \ldots, j_p$ from Procedure 2, we compute W as follows. Let $W^{(1)}, \ldots, W^{(\tau)}$ be matrices such that $W^{(\ell)}(\cdot, t) = S^{(\ell)}(\cdot, j_t)/\sqrt{pP'_{j_t}}$ for all $t \in [p]$ and $\ell \in [\tau]$, where $P'_j = \frac{1}{p}\sum_{t=1}^{p} Q_{j|i_t}$ for $j \in [n]$. Define the matrix W as

$$W = W^{(1)} + \cdots + W^{(\tau)}. \tag{77}$$

Algorithm 4: Approximation of left singular vector space.

---

input : Set of m-by-n matrices $\{A^{(1)}, \ldots, A^{(\tau)}\}$ with $\mathrm{S}(A^{(\ell)})$ and $\mathrm{Q}(A^{(\ell)})$ for $\ell \in [\tau]$; threshold σ; error parameter ϵ, η.

1 Set $p = \Theta(\frac{\tau^6(\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2)^4}{\epsilon^4\sigma^8\eta^2})$;

2 Use Procedure 2 to obtain row indices $i_1, \ldots, i_p$ and probability distribution $\{P_1, \ldots, P_m\}$; Let $S^{(1)}, \ldots, S^{(\tau)}$ be matrices such that $S^{(\ell)}(t, \cdot) = A^{(\ell)}(i_t, \cdot)/\sqrt{pP_{i_t}}$ for all $t \in [p]$ and $\ell \in [\tau]$. Let $S = S^{(1)} + \cdots + S^{(\tau)}$;

3 Use Procedure 3 to obtain column indices $j_1, \ldots, j_p$ and probability distributions $\{Q_{1|i_t}, \ldots, Q_{n|i_t}\}$;

4 Let $W^{(1)}, \ldots, W^{(\tau)}$ be matrices such that $W^{(\ell)}(\cdot, t) = S^{(\ell)}(\cdot, j_t)/\sqrt{pP'_{j_t}}$ for all $t \in [p]$ and $\ell \in [\tau]$, where $P'_j = \frac{1}{p}\sum_{t=1}^{p} Q_{j|i_t}$ for $j \in [n]$. Let $W = W^{(1)} + \cdots + W^{(\tau)}$;

5 Compute the top r singular values $\sigma_1, \ldots, \sigma_r$ of W and their corresponding left singular vectors $u_1, \ldots, u_r$ such that $\sigma_r \geq \sigma > \sigma_{r+1}$;

6 Output $\sigma_1, \ldots, \sigma_r$, $u_1, \ldots, u_r$, and $i_1, \ldots, i_p$;

With the weighted sampling method, we obtained a small submatrix W from A. Now, we use the singular values and singular vectors of W to approximate the ones of A. This is shown in Algorithm 4, which is similar to [Tan19, Algorithm 2].

The output of Algorithm 4 characterizes a matrix $V \in \mathbb{C}^{n\times r}$ defined as.

$$V(\cdot, j) := \frac{S^\dagger}{\sigma_j} u_j, \tag{78}$$

for $j \in [r]$, where S is defined by $i_1, \ldots, i_p$ and A as in Eq. (76).

Algorithm 4 is similar to the main algorithm in [FKV04] except for the different sampling method used here. In terms of the low-rank approximation, a similar result holds as follows.

Lemma 5.3. *Let $A = A^{(1)} + \cdots + A^{(\tau)}$. Let $\epsilon, \eta > 0$ be the error parameter. Take $\mathrm{SQ}(A^{(\ell)})$, $\sigma > 0$, ϵ, and η as the input of Algorithm 4 to obtain $\sigma_1, \ldots, \sigma_r$, $u_1, \ldots, u_r$, and $i_1, \ldots, i_p$. Let $V \in \mathbb{C}^{n\times r}$ be defined as in Eq. (78). Then, with probability at least $9/10$, it holds that $\|AVV^\dagger - A\|_F^2 \leq \|A - A_r\|_F^2 + \eta\epsilon^2 \sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F$, where A_r is the best rank-r approximation of A. Moreover, r satisfies $\ell_A(\sigma(1+\eta)) \leq r \leq \ell_A(\sigma(1-\eta))$.*

Corollary 5.4. *With probability at least $9/10$, it holds that $\|AVV^\dagger - A_{\sigma,\eta}\|_F \leq \epsilon\sqrt{\sum_\ell \|A^{(\ell)}\|_F^2}/\sqrt{\eta}$.*

An important result of Algorithm 4 is that the matrix formed by the vectors $u_1, \ldots, u_r$ is approximately an isometry, as stated in the following lemma:

Lemma 5.5 (Approximate orthogonality)**.** *Let $A = A^{(1)} + \cdots + A^{(\tau)}$. Let $\epsilon, \eta > 0$ be the error parameter. Take $\mathrm{SQ}(A^{(\ell)})$, $\sigma > 0$, ϵ, and η as the input of Algorithm 4 to obtain $\sigma_1, \ldots, \sigma_r$, $u_1, \ldots, u_r$, and $i_1, \ldots, i_p$. Let $V \in \mathbb{C}^{n\times r}$ be defined as in Eq. (78). Then, with probability at least $9/10$, V is $\mathcal{O}(\eta\epsilon^2/\tau)$-approximate isometry.*

Remark 5.6 (Succinct description)**.** We denote the output $\sigma_1, \ldots, \sigma_r$, $u_1, \ldots, u_r$, and $i_1, \ldots, i_p$ of Algorithm 4 as the **succinct description** of V. We can obtain V from the succinct description via Eq. (78).

In the following, we show that given the succinct description of V as in Algorithm 4, we can approximately implement $\mathrm{SQ}(V)$.

Lemma 5.7. *Given the succinct description of V, with probability at least 9/10, we can obtain $\mathrm{SQ}^{\eta\epsilon^2/\tau}_{\eta\epsilon^2/\tau}(V)$ with $\mathbf{q}(V) = \mathcal{O}\Big(\frac{\tau^7(\sum_\ell \|A^{(\ell)}\|_F^2)^4}{\epsilon^4\sigma^8\eta^2}\Big)$, $\mathbf{s}(V) = \mathcal{O}\Big(\frac{\tau^{13}(\sum_\ell \|A^{(\ell)}\|_F^2)^9}{\epsilon^8\sigma^{18}\eta^4}\Big)$, $\mathbf{n}^{\eta\epsilon^2/\tau}_{\eta\epsilon^2/\tau}(V) = \widetilde{\mathcal{O}}(1)$.*

The proofs of the lemmas above are postponed in Appendix A.

By using Remark 3.5, we have the following.

Lemma 5.8. *Let $V \in \mathbb{C}^{n\times r}$, $U \in \mathbb{C}^{n\times r}$ and $A = \sum_\ell^\tau A^{(\ell)} \in \mathbb{C}^{n\times n}$ be a Hermitian matrix. Given $\mathrm{SQ}(A^{(\ell)})$ for $\ell \in [\tau]$, and $\mathrm{Q}(V)$ and $\mathrm{Q}(U)$, one can output a Hermitian matrix $\tilde{B} \in \mathbb{C}^{r\times r}$ such that $\|U^\dagger AV - \tilde{B}\|_F \le \epsilon_s$ with probability $1-\delta$ by using $\widetilde{\mathcal{O}}\Big((p+\log n)\frac{2^{16}r^9\tau^3}{\epsilon^2}\log\frac{1}{\delta}\Big)$ samples and time.*

Proof. Let $B_\ell = U^\dagger A^{(\ell)} V$ for $\ell \in [\tau]$ and $B = \sum_{\ell=1}^\tau B_\ell$. $B_\ell(i,j) = U^\dagger(i,\cdot)A^{(\ell)}V(\cdot,j)$. Then, we can estimate $U^\dagger(i,\cdot)A^{(\ell)}V(\cdot,j) = \mathrm{Tr}[A^{(\ell)}(V(\cdot,j)U^\dagger(i,\cdot))]$ by Remark 3.5, which gives error at most $\epsilon_s/r\sqrt{\tau}$ with probability $1-\frac{\delta}{\tau r^2}$ by using $\widetilde{\mathcal{O}}\Big(\|A^{(\ell)}\|_F\|U(i,\cdot)\|\|V(j,\cdot)\|\frac{r^2\tau}{\epsilon_s^2}\log\frac{r^2\tau}{\delta}\Big)$ queries. We denote the estimation to $B_\ell(i,j)$ as $\tilde{B}_\ell(i,j)$. Therefore, for a single B_ℓ, we have

$$\Pr\left[\bigwedge_{i,j\in[r]}(|B_\ell(i,j)-\tilde{B}_\ell(i,j)| \le \epsilon_s/r\sqrt{\tau})\right] \ge 1-\delta/\tau. \tag{79}$$

Then, for $B_1, \ldots, B_\tau$,

$$\Pr\left[\bigwedge_{i,j\in[r]}\bigwedge_{\ell\in[\tau]}(|B_\ell(i,j)-\tilde{B}_\ell(i,j)| \le \epsilon_s/r\sqrt{\tau})\right] \ge 1-\delta. \tag{80}$$

Now, we are guaranteed that for all $\ell \in [\tau]$, $-\epsilon\vec{1}\vec{1}^\dagger \le B_\ell - \tilde{B}_\ell \le \epsilon\vec{1}\vec{1}^\dagger$ with probability at least $1-\delta$. Let $\tilde{B} = \sum_\ell \tilde{B}_\ell$. With probability $1-\delta$,

$$\|B-\tilde{B}\|_F \le \sqrt{r^2\tau(\epsilon_s^2/r^2\tau)} = \epsilon_s. \tag{81}$$

$\square$

5.2 Proofs of Theorem 5.1 and Theorem 5.2

We prove Theorem 5.1 and Theorem 5.2 in this subsection.

Proof of Theorem 5.1. We use Algorithm 5, which outputs succinct descriptions of $\check{U}$ and $\check{V}$, and the matrix D. The time complexity of Algorithm 5 is $O(\frac{\tau^{18}(\sum_\ell\|A^{(\ell)}\|_F^2)^{12}}{\epsilon^{12}\sigma^{24}\eta^6})$, because the SVD in Algorithm 4 takes $O(p^3)$ time. Given succinct description of $\check{U}$ and $\check{V}$, we can get $\mathrm{SQ}^{\eta\epsilon^2/\tau}_{\eta\epsilon^2/\tau}(\check{V})$ and $\mathrm{SQ}^{\eta\epsilon^2/\tau}_{\eta\epsilon^2/\tau}(\check{U})$ using Lemma 5.7.

The fact that $\check{U}$, $\check{V}$, and D satisfy statements 1 and 2 follows from Corollary 5.4. $\square$

Algorithm 5: Succinct description of the singular value decomposition of A.

input : $A = A^{(1)} + \cdots + A^{(\tau)}$ with $\mathrm{SQ}(A^{(\ell)})$ and $\mathrm{SQ}(A^{(\ell)^\dagger})$ for $\ell \in [\tau]$; error parameter ϵ.

Compute succinct description of V by calling Algorithm 4 with $\{A^{(\ell)}\}$ and ϵ;

Compute succinct description of U by calling Algorithm 4 with $\{A_\ell^\dagger\}$ and ϵ;

Compute the matrix $U^\dagger AV$ according to Lemma 5.8;

Compute the singular value decomposition $U'DV'^\dagger$ of the matrix $U^\dagger AV$;

Output U' and V', D, and the succinct descriptions of U and V.

Proof of Theorem 5.2. We use Algorithm 6, which outputs a succinct description of $\check{U}$ and the matrix D. The time complexity of Algorithm 6 is $O(\frac{\tau^{18}(\sum_\ell \|A^{(\ell)}\|_F^2)^{12}}{\epsilon^{12}\sigma^{24}\eta^6})$, because the SVD in Algorithm 4 takes $O(p^3)$ time. Given succinct description of $\check{U}$, we can get $SQ(\check{U}^\dagger)$ using Lemma 5.7.

Algorithm 6: Approximation of the spectral decomposition of A.

input : $A = A^{(1)} + \cdots + A^{(\tau)}$ with with $\mathrm{SQ}(A^{(\ell)})$ and $\mathrm{SQ}(A^{(\ell)^\dagger})$ for $\ell \in [\tau]$; error parameter ϵ.

Compute succinct description of U by calling Algorithm 4 with $\{A_\ell^\dagger\}$ and ϵ;

Compute the matrix $U^\dagger AU$ according to Lemma 5.8;

Compute the spectral decomposition $U'DU'^\dagger$ of the matrix $U^\dagger AU$;

Output U', D, and succinct description of U.

The fact that $\check{U}$ and D satisfy statements 1 and 2 follows from Corollary 5.4. ☐

6 Singular value transformation and matrix functions

In this section, we show how to apply any smooth function f to the singular values of A efficiently when we have SQ access (Definition 2.6) to both A and $A^\dagger$.

We first introduce some lemmas which we will use in the section.

Lemma 6.1 ([FN09, Theorem 10])**.** *Let $f\colon [-R, R] \to \mathbb{C}$ be L-Lipschitz continuous and A, B be Hermitian matrices with $\|A\| \le R$, $\|B\| \le R$, and $\|A - B\| \le \epsilon$. Then it holds that*

$$\|f(A) - f(B)\| \le 4L\epsilon\left(\log\left(1 + \frac{2R}{\epsilon}\right) + 1\right)^2. \tag{82}$$

Lemma 6.2 ([GSLW19, Corollary 21])**.** *Let $f\colon [0, R] \to \mathbb{C}$ be L-Lipschitz continuous and A, B be Hermitian matrices $\|A\| \le R$, $\|B\| \le R$, and $\|A - B\| \le \epsilon$. Then it holds that*

$$\left\|f^{(\mathrm{SV})}(A) - f^{(\mathrm{SV})}(B)\right\| \le 8L\epsilon\left(\log\left(1 + \frac{R}{\epsilon}\right) + 1\right)^2. \tag{83}$$

6.1 Approximating singular value functions

Theorem 3.1 (Singular value transformation)**.** *Let $A = A^{(1)} + \cdots + A^{(\tau)} \in \mathbb{C}^{m\times n}$. For $\epsilon \in (0, L)$, $\nu = \widetilde{\mathcal{O}}\big(\epsilon^2/(\alpha^2 \sum_\ell \|A^{(\ell)}\|_F^2)\big)$ there exists an algorithm with time complexity $\widetilde{\mathcal{O}}\Big(\Big(\frac{\tau L^2(\sum_\ell \|A^{(\ell)}\|_F^2)}{\epsilon^2}\Big)^{18}\mathbf{sq}(A^{[\tau]^{(\dagger)}})\Big)$*

which provides succinct descriptions of $\widetilde{\mathcal{O}}\Big(\frac{\epsilon^2}{L^2\|A\|_F^2}\Big)$*-approximate isometries* $\check{V}$, $\check{U}$, *and a diagonal matrix* $D \in \mathbb{R}^{r\times r}$ *such that the matrix* $B := \check{U}D\check{V}^\dagger$ *satisfies* $\big\|B - f^{(\mathrm{SV})}(A)\big\| \le \epsilon$ *with probability* $9/10$. *Moreover, the provided representation of* B *enables* $\mathrm{SQ}_\nu^\nu(B)$ *access in time* $\mathbf{s}(B) = \mathcal{O}\big(r^2\mathbf{sq}(\check{U},\check{V})\big)$, $\mathbf{q}(B) = \mathcal{O}\big(r\mathbf{sq}(\check{U},\check{V})\big)$, $\mathbf{n}_\nu(B) = \mathbf{n}^\nu(B) = \mathcal{O}\big(r^2/\nu^2\mathbf{sq}(\check{U},\check{V})\big)$, *where* $r \le \ell_{\frac{A}{\|A\|_F}}(\widetilde{\Theta}(\frac{\epsilon}{L\|A\|_F})) = \widetilde{\mathcal{O}}\Big(\frac{L^2\|A\|_F^2}{\epsilon^2}\Big)$ *and* $\mathbf{sq}(\check{U},\check{V}) = \widetilde{\mathcal{O}}\bigg(\Big(\frac{\tau L^2(\sum_\ell\|A^{(\ell)}\|_F^2)}{\epsilon^2}\Big)^{13}\mathbf{sq}(A^{[\tau]}{}^{(\dagger)})\bigg)$.

Proof. The algorithm is very simple, we use Theorem 5.1 to obtain an approximate singular value decomposition such that $\|A_{\frac{\sigma}{2},\frac{1}{2}} - \check{U}D\check{V}^\dagger\|_F \le \xi$, where $D \in \mathbb{R}_+^{r\times r}$ is a diagonal matrix and $\check{U} \in \mathbb{C}^{r\times m}, \check{V} \in \mathbb{C}^{r\times n}$ are approximate isometries such that $\big\|\check{U} - U\big\| \le \nu \le \frac{1}{6}$ and $\big\|\check{V} - V\big\| \le \nu$ for some isometries U and V, and we get $\mathrm{SQ}_\nu^{r\nu}(\check{U})$, $\mathrm{SQ}_\nu^{r\nu}(\check{V})$ access. Then we simply apply the function to D providing a succinct representation of B.

Using this representation we can easily get sample and query access to the i-th row of B: just query the i-th row of $\check{U}$ then multiply with $f(D)$. Use the resulting vector as w in Proposition 3.6. This requires $\mathcal{O}(r)$ queries to $\check{U}$ and D, and the complexity follows from Proposition 3.6, using the observation that $C = \Theta(1)$ due to the fact that $\check{V}$ is an approximate isometry. Thus querying a matrix entry takes $\mathcal{O}(r)$ queries, and taking a sample from a row takes $\mathcal{O}\big(r^2\big)$ queries, while estimating a row norm to $\mathcal{O}(\nu)$ multiplicative error takes $\mathcal{O}\big(r^2/\nu^2\big)$ queries.

Now we solve the problem of sampling a row index by sampling an entry from B itself, then keeping the row index and ignoring the column index. Observe that B is a sum of r dyadic matrices that are approximately orthogonal according to the Hilbert-Schmidt (trace) inner product. Considering the dyadic matrices as flat vectors we can again apply Proposition 3.6. Due to the approximate orthogonality we get once again that $C = \Theta(1)$, thus taking a row-index sample takes $\mathcal{O}\big(r^2\big)$ queries, while estimating the Frobenius norm to $\mathcal{O}(\nu)$ multiplicative error takes $\mathcal{O}\big(r^2/\nu^2\big)$ queries.

We finish the proof by showing the correctness of the algorithm. In the proof we will write $\|\cdot\|_{(F)}$ if a statement is valid for both norms $\|\cdot\|_F$ and $\|\cdot\|$. Using this notation we have that $\Big\|A_{\frac{\sigma}{2},1}\Big\|_{(F)} \le \|A\|_{(F)}$, $\Big\|A_{\frac{\sigma}{2},1} - A\Big\|_{(F)} \le \|A\|_{(F)}$ and $\Big\|A_{\frac{\sigma}{2},1} - A\Big\| \le \sigma$. Now we use the well-known property that $\|XY\|_{(F)} \le \|X\|\|Y\|_{(F)}$. It follows that $\|\check{U}D\check{V}^\dagger - UDV^\dagger\|_{(F)} \le (2+\nu)\nu\|D\|_{(F)} \le \frac{5}{2}\nu\|D\|_{(F)}$. Therefore $\|D\|_{(F)} = \big\|UDV^\dagger\big\|_{(F)} \le \|UDV^\dagger - \check{U}D\check{V}^\dagger\|_{(F)} + \|\check{U}D\check{V}^\dagger - A_{\frac{\sigma}{2},1}\|_{(F)} + \Big\|A_{\frac{\sigma}{2},1} - A\Big\|_{(F)} \le \frac{5}{2}\nu\|D\|_{(F)} + \xi + \|A\|_{(F)}$ and so $\|D\|_{(F)} \le 2(\|A\|_{(F)} + \sigma)$. Now we get $\|\check{U}D\check{V}^\dagger - UDV^\dagger\|_{(F)} \le 5\nu(\|A\|_{(F)} + \xi)$, and thus also $\|A - UDV^\dagger\| \le \Big\|A - A_{\frac{\sigma}{2},1}\Big\| + \|A_{\frac{\sigma}{2},1} - \check{U}D\check{V}^\dagger\| + \|UDV^\dagger - \check{U}D\check{V}^\dagger\| \le \sigma + \xi + 5\nu(\|A\| + \xi)$.

By Lemma 6.2 it follows that

$$\|f^{(\mathrm{SV})}(A) - Uf(D)V^\dagger\| \le 8L[\sigma + \xi + 5\nu(\|A\| + \xi)]\left(\log\left(1 + \frac{2(\|A\| + \sigma)}{[\sigma + \xi + 5\nu(\|A\| + \xi)]}\right) + 1\right)^2. \quad (84)$$

Of course we also have $\|Uf(D)V^\dagger\| \le \|f^{(\mathrm{SV})}(A)\| + \|f^{(\mathrm{SV})}(A) - Uf(D)V^\dagger\|$, and thus also $\|Uf(D)V^\dagger - \check{U}f(D)\check{V}^\dagger\| \le \frac{5}{2}\nu(\|f^{(\mathrm{SV})}(A)\| + \|f^{(\mathrm{SV})}(A) - Uf(D)V^\dagger\|) \le \frac{5}{2}\nu L\|A\| + \frac{5}{12}\|f^{(\mathrm{SV})}(A) - Uf(D)V^\dagger\|$. So we obtain

$$\|f^{(\mathrm{SV})}(A) - \check{U}f(D)\check{V}^\dagger\| \leq \|f^{(\mathrm{SV})}(A) - Uf(D)V^\dagger\| + \|Uf(D)V^\dagger - \check{U}f(D)\check{V}^\dagger\| \tag{85}$$

$$\leq 12L[\sigma+\xi+6\nu(\|A\|+\xi)]\left(\log\left(1+\frac{2(\|A\|+\sigma)}{[\sigma+\xi+5\nu(\|A\|+\xi)]}\right)+1\right)^2 \tag{86}$$

$$\leq 12L[\sigma+2\xi+6\nu\|A\|]\left(\log\left(1+\frac{2(\|A\|+\sigma)}{[\sigma+\xi+5\nu\|A\|]}\right)+1\right)^2. \tag{87}$$

Let $c = \left(\log\left(1+\frac{L\|A\|}{\epsilon'}\right)+1\right)^{-2}$, choosing $\sigma = \mathcal{O}(\epsilon'/(cL))$, $\xi = \mathcal{O}(\epsilon'/(cL))$, and $\nu = \mathcal{O}(\epsilon'/(cL\|A\|))$ with appropriate constants we get that $\|f^{(\mathrm{SV})}(A) - \check{U}f(D)\check{V}^\dagger\| \leq \epsilon'$. Recall that from Theorem 5.1 we get guarantees $\xi = \epsilon\sqrt{\sum_\ell \|A^{(\ell)}\|_F^2}$ and $\nu = \mathcal{O}\big(\epsilon^2/\tau\big)$. Due to the Cauchy-Schwarz inequality we have that $\tau\sum_\ell \|A^{(\ell)}\|_F^2 \geq \|\sum_\ell A^{(\ell)}\|_F^2 = \|A\|_F^2 \geq \|A\|^2$, and we see that choosing $\epsilon = \mathcal{O}\Big(\epsilon'/\Big(cL\sqrt{\sum_\ell \|A^{(\ell)}\|_F^2}\Big)\Big)$ suffices. Since the complexity of obtaining the succinct representation in Theorem 5.1 is $\mathcal{O}\Big(\frac{\tau^{18}(\sum_\ell \|A^{(\ell)}\|_F^2)^{12}}{\epsilon^{12}\sigma^{24}\eta^6}\Big)$, the complexity of obtaining B is

$$\widetilde{\mathcal{O}}\left(\left(\frac{\tau L^2(\sum_\ell \|A^{(\ell)}\|_F^2)}{\epsilon'^2}\right)^{18}\right). \qquad \square$$

6.2 Approximating matrix functions

Theorem 3.3 (Matrix function). *Let $A = \gamma I + A^{(1)} + \cdots + A^{(\tau)}$ where for each $A^{(\ell)} \in \mathbb{C}^{n\times n}$ is Hermitian. For $\epsilon \in (0, L)$, $\nu = \widetilde{\mathcal{O}}\big(\epsilon^2/(L^2\sum_\ell \|A^{(\ell)}\|_F^2)\big)$ there exists an algorithm with time complexity $\widetilde{\mathcal{O}}\bigg(\Big(\frac{\tau L^2(\sum_\ell \|A^{(\ell)}\|_F^2)}{\epsilon^2}\Big)^{18}\mathbf{sq}(A^{[\tau]})\bigg)$ which provides a succinct description of a $\widetilde{\mathcal{O}}\Big(\frac{\epsilon^2}{L^2\|A\|_F^2}\Big)$-approximate isometry $\check{U}$, and a diagonal matrix $D \in \mathbb{R}^{r\times r}$ such that the matrix $B := \check{U}D\check{U}^\dagger$ satisfies $\|f(\gamma)I + B - f(A)\| \leq \epsilon$ with probability $9/10$. Moreover, the provided representation of B enables $\mathrm{SQ}_\nu^\nu(B)$ access in time $\mathbf{s}(B) = \mathcal{O}\big(r^2\mathbf{sq}(\check{U})\big)$, $\mathbf{q}(B) = \mathcal{O}\big(r\mathbf{sq}(\check{U})\big)$, $\mathbf{n}_\nu(B) = \mathbf{n}^\nu(B) = \mathcal{O}\big(r^2/\nu^2\mathbf{sq}(\check{U})\big)$, where $r \leq \ell_{\frac{A}{\|A\|_F}}(\widetilde{\Theta}(\frac{\epsilon}{L\|A\|_F})) = \widetilde{\mathcal{O}}\Big(\frac{L^2\|A\|_F^2}{\epsilon^2}\Big)$ and $\mathbf{sq}(\check{U}) = \widetilde{\mathcal{O}}\bigg(\Big(\frac{\tau L^2(\sum_\ell \|A^{(\ell)}\|_F^2)}{\epsilon^2}\Big)^{13}\mathbf{sq}(A^{[\tau]})\bigg)$.*

Proof. The proof is analogous to the proof of Theorem 3.1, just with respect to the function $g(x) := f(\gamma + x) - f(\gamma)$. $\square$

References

[Aar15] Scott Aaronson. Read the fine print. *Nature Physics*, 11(4):291, 2015.

[ADBL19] Juan Miguel Arrazola, Alain Delgado, Bhaskar Roy Bardhan, and Seth Lloyd. Quantum-inspired algorithms in practice, 2019, arXiv:1905.10415.

[AK07] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 227–236. ACM, 2007.

[ATS03] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 20–29. ACM, 2003, arXiv:quant-ph/0301023.

[BCC+14] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Exponential improvement in precision for simulating sparse Hamiltonians. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 283–292, 2014, arXiv:1312.1414.

[BCC+15] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Simulating Hamiltonian dynamics with a truncated Taylor series. *Physical Review Letters*, 114(9):090502, 2015, arXiv:1412.4687.

[BCK15] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science*, pages 792–809, 2015, arXiv:1501.01715.

[BKL+19] Fernando G. S. L. Brandão, Amir Kalev, Tongyang Li, Cedric Yen-Yu Lin, Krysta M. Svore, and Xiaodi Wu. Quantum SDP solvers: Large speed-ups, optimality, and applications to quantum learning. In *46th International Colloquium on Automata, Languages, and Programming*, volume 132, page 27. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, arXiv:1710.02581.

[CD16] Iris Cong and Luming Duan. Quantum discriminant analysis for dimensionality reduction and classification. *New Journal of Physics*, 18(7):073011, 2016, arXiv:1510.00113.

[CLLW19] Nai-Hui Chia, Tongyang Li, Han-Hsuan Lin, and Chunhao Wang. Quantum-inspired classical sublinear-time algorithm for solving low-rank semidefinite programming via sampling approaches, 2019, arXiv:1901.03254.

[CLS+19] Zhihuai Chen, Yinan Li, Xiaoming Sun, Pei Yuan, and Jialin Zhang. A quantum-inspired classical algorithm for separable non-negative matrix factorization, 2019, arXiv:1907.05568.

[CLW18] Nai-Hui Chia, Han-Hsuan Lin, and Chunhao Wang. Quantum-inspired sublinear classical algorithms for solving low-rank linear systems, 2018, arXiv:1811.04852.

[DBH19] Chen Ding, Tian-Yi Bao, and He-Liang Huang. Quantum-inspired support vector machine, 2019, arXiv:1906.08902.

[DHLT19] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao. A quantum-inspired algorithm for general minimum conical hull problems, 2019, arXiv:1907.06814.

[DKM06] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006.

[DKR02] Petros Drineas, Iordanis Kerenidis, and Prabhakar Raghavan. Competitive recommendation systems. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 82–90, 2002.

[DM07] Petros Drineas and Michael W. Mahoney. A randomized algorithm for a tensor-based generalization of the singular value decomposition. *Linear Algebra and its Applications*, 420(2-3):553–571, 2007.

[Fey82] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488, 1982.

[FKV04] Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004.

[FN09] Yu. B. Farforovskaya and L. Nikolskaya. Modulus of continuity of operator functions. *St. Petersburg Mathematical Journal*, 20(3):493–506, 2009.

[GLT18] András Gilyén, Seth Lloyd, and Ewin Tang. Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension, 2018, arXiv:1811.04909.

[GR02] Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*, 2002.

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 193–204. ACM, 2019, arXiv:1806.01838.

[HHL09] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009, arXiv:0811.3171.

[Kal07] Satyen Kale. *Efficient algorithms using the multiplicative weights update method*. PhD thesis, Princeton University, 2007.

[KP17] Iordanis Kerenidis and Anupam Prakash. Quantum recommendation systems. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference*, pages 49:1–49:21, 2017, arXiv:1603.08675.

[KV17] Ravindran Kannan and Santosh Vempala. Randomized algorithms in numerical linear algebra. *Acta Numerica*, 26:95–135, 2017.

[LC17] Guang Hao Low and Isaac L. Chuang. Optimal Hamiltonian simulation by quantum signal processing. *Physical Review Letters*, 118(1):010501, 2017, arXiv:1606.02685.

[LGZ16] Seth Lloyd, Silvano Garnerone, and Paolo Zanardi. Quantum algorithms for topological and geometric analysis of data. *Nature Communications*, 7:10138, 2016, arXiv:1408.3106.

[Llo96] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073, 1996.

[LMR13] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning, 2013, arXiv:1307.0411.

[LMR14] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631, 2014, arXiv:1307.0401.

[LRS15] James R. Lee, Prasad Raghavendra, and David Steurer. Lower bounds on the size of semidefinite programming relaxations. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing*. ACM, 2015, arXiv:1411.6317.

[Mah11] Michael W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011, arXiv:1104.5557.

[McD89] Colin McDiarmid. *On the method of bounded differences*, page 148–188. London Mathematical Society Lecture Note Series. Cambridge University Press, 1989.

[MMD08] Michael W. Mahoney, Mauro Maggioni, and Petros Drineas. Tensor-CUR decompositions for tensor-based data. *SIAM Journal on Matrix Analysis and Applications*, 30(3):957–987, 2008.

[Pra14] Anupam Prakash. *Quantum algorithms for linear algebra and machine learning*. PhD thesis, UC Berkeley, 2014.

[Pre18] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018, arXiv:1801.00862.

[RML14] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical Review Letters*, 113(13):130503, 2014, arXiv:1307.0471.

[RSML18] Patrick Rebentrost, Adrian Steffens, Iman Marvian, and Seth Lloyd. Quantum singular-value decomposition of nonsparse low-rank matrices. *Physical review A*, 97(1):012327, 2018.

[RWC+18] Alessandro Rudi, Leonard Wossnig, Carlo Ciliberto, Andrea Rocchetto, Massimiliano Pontil, and Simone Severini. Approximating Hamiltonian dynamics with the Nyström method, 2018, arXiv:1804.02484.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997, arXiv:quant-ph/9508027. Earlier version in FOCS'94.

[Tan18] Ewin Tang. Quantum-inspired classical algorithms for principal component analysis and supervised clustering, 2018, arXiv:1811.00414.

[Tan19] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 217–228. ACM, 2019, arXiv:1807.04271.

[WS01] Christopher K.I. Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, pages 682–688, 2001.

[ZFF19] Zhikuan Zhao, Jack K. Fitzsimons, and Joseph F. Fitzsimons. Quantum-assisted Gaussian process regression. *Physics Review A*, 99:052331, May 2019, arXiv:1512.03929.

A Proofs of the subsampling techniques

The intuition of the result of the Algorithm 4 is that $S^\dagger S \approx A^\dagger A$ and $SS^\dagger \approx WW^\dagger$. To see this, we first prove the following technical lemma relates the three quantities: $\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2$, $\sum_{\ell=1}^{\tau}\|S^{(\ell)}\|_F^2$, and $\sum_{\ell=1}^{\tau}\|W^{(\ell)}\|_F^2$, which is an extension of [FKV04, Lemma 1].

Lemma A.1. *Let $A = A^{(1)} + \cdots + A^{(\tau)}$ be a matrix with the sampling access for each $A^{(\ell)}$ as in Definition 2.6. Let S and W be defined by Eqs. (76) and (77). Then, with probability at least $1 - 2\tau^2/p$ it holds that*

$$\frac{1}{\tau+1}\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2 \le \sum_{\ell=1}^{\tau}\|S^{(\ell)}\|_F^2 \le \frac{2\tau+1}{\tau+1}\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2, \tag{88}$$

and

$$\frac{1}{\tau+1}\sum_{\ell=1}^{\tau}\|S^{(\ell)}\|_F^2 \le \sum_{\ell=1}^{\tau}\|W^{(\ell)}\|_F^2 \le \frac{2\tau+1}{\tau+1}\sum_{\ell=1}^{\tau}\|S^{(\ell)}\|_F^2, \tag{89}$$

Proof. We first evaluate $\mathbb{E}(\|S^{(\ell)}\|_F^2)$ as follows. For all $\ell \in [\tau]$,

$$\mathbb{E}\Big(\|S^{(\ell)}\|_F^2\Big) = \sum_{i=1}^{p}\mathbb{E}\Big(\Big\|S^{(\ell)}(i,\cdot)\Big\|^2\Big) = \sum_{i=1}^{p}\sum_{j=1}^{m}P_j\frac{\|A^{(\ell)}(j,\cdot)\|^2}{pP_j} = \sum_{j=1}^{m}\Big\|A^{(\ell)}(j,\cdot)\Big\|^2 = \|A^{(\ell)}\|_F^2. \tag{90}$$

Then we have

$$\Big\|S^{(\ell)}(i,\cdot)\Big\|^2 = \sum_{j=1}^{n}\frac{|A^{(\ell)}(i,j)|^2}{pP_i} \le \sum_{j=1}^{n}\frac{2|A^{(\ell)}(i,j)|^2\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2}{p\sum_{j=1}^{\tau}\|A^(j)(i,\cdot)\|^2} \tag{91}$$

$$= \frac{2\|A^{(\ell)}(i,\cdot)\|^2\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2}{p\sum_{j=1}^{\tau}\|A^{(j)}(i,\cdot)\|^2} \le \frac{2\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2}{p}. \tag{92}$$

Note that the quantity $\|S^{(\ell)}\|_F^2$ can be viewed as a sum of p independent random variables $\|S^{(\ell)}(1,\cdot)\|^2, \ldots, \|S^{(\ell)}(p,\cdot)\|^2$. As a result,

$$\mathrm{Var}(\|S^{(\ell)}\|_F^2) = p\mathrm{Var}(\Big\|S^{(\ell)}(i,\cdot)\Big\|^2) \le p\mathbb{E}\Big(\Big\|S^{(\ell)}(i,\cdot)\Big\|^4\Big) \tag{93}$$

$$\le p\sum_{i=1}^{m}P_i\left(\frac{2\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2}{p}\right)^2 = \frac{2\Big(\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2\Big)^2}{p}. \tag{94}$$

According to Chebyshev's inequality, we have

$$\Pr\left(\Big|\|S^{(\ell)}\|_F^2 - \|A^{(\ell)}\|_F^2\Big| \ge \frac{\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2}{\tau}\right) \le \frac{2\Big(\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2\Big)^2}{p} = \frac{2\tau^2}{p}. \tag{95}$$

Therefore, with probability at least $1-\frac{2\tau^2}{p}$, it holds that

$$-\frac{1}{\tau+1}\sum_{j\neq\ell}\|A^{(j)}\|_F^2+\frac{\tau}{\tau+1}\|A^{(\ell)}\|_F^2\leq\|S^{(\ell)}\|_F^2\leq\frac{1}{\tau+1}\sum_{j\neq\ell}\|A^{(j)}\|_F^2+\frac{\tau+2}{\tau+1}\|A^{(\ell)}\|_F^2, \tag{96}$$

which implies that

$$\frac{1}{\tau+1}\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2\leq\sum_{\ell=1}^{\tau}\|S^{(\ell)}\|_F^2\leq\frac{2\tau+1}{\tau+1}\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2. \tag{97}$$

Eq. (89) can be proven in a similar way. □

Now, we show the that $A^\dagger A\approx S^\dagger S$ and $WW^\dagger\approx SS^\dagger$ as the following lemma.

Lemma A.2. *Let $A=A^{(1)}+\cdots+A^{(\tau)}$ be a matrix with the sampling access for each $A^{(\ell)}$ as in Definition 2.6. Let S and W be defined by Eqs. (76) and (77). Letting $\theta=\tau\sqrt{\frac{100}{p}}$, then, with probability at least $9/10$, the following holds:*

$$\left\|A^\dagger A-S^\dagger S\right\|_F\leq\theta\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2,\ \textit{and} \tag{98}$$

$$\left\|SS^\dagger-WW^\dagger\right\|_F\leq\theta\sum_{\ell=1}^{\tau}\|S^{(\ell)}\|_F^2\leq 2\theta\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2. \tag{99}$$

Proof. This lemma is an adaption of [FKV04, Lemma 2] to our weighted sampling framework. We first show that the sampling probabilities P_i and $Q_{j|i_t}$ satisfy Assumptions 1 and 2 in [FKV04].

$$\begin{aligned}
P_i&=\sum_{j=1}^{\tau}\mathcal{D}_{\mathrm{rows}(A^{(j)})}(i)\|A^{(j)}\|_F^2/\left(\sum_{\ell=1}^{\tau}\|A^{(\ell)}\|_F^2\right) &(100)\\
&=\frac{\sum_\ell\|A^{(\ell)}(i,\cdot)\|^2}{\sum_\ell\|A^{(\ell)}\|_F^2} &(101)\\
&\geq\frac{\|\sum_\ell A^{(\ell)}(i,\cdot)\|_F^2}{\tau\sum_\ell\|A^{(\ell)}\|_F^2} &(102)\\
&\geq\left(\frac{\|A\|_F^2}{\tau\sum_\ell\|A^{(\ell)}\|_F^2}\right)\frac{\|A(i,\cdot)\|^2}{\|A\|_F^2}, &(103)
\end{aligned}$$

and

$$\begin{aligned}
Q_{j|i_t}\frac{P_i}{P_{i,j}}&=\sum_{k=1}^{\tau}\mathcal{D}_{A^{(k)}(i_t,\cdot)}(j)\left\|A^{(k)}(i_t,\cdot)\right\|^2/\left(\sum_{\ell=1}^{\tau}\left\|A^{(\ell)}(i_t,\cdot)\right\|^2\right)\frac{P_i}{P_{i,j}} &(104)\\
&=\frac{\sum_{\ell=1}^{\tau}|A^{(\ell)}(i,j)|^2}{\sum_{\ell=1}^{\tau}\|A^{(\ell)}(i,\cdot)\|^2}\left(\frac{\sum_\ell\|A^{(\ell)}(i,\cdot)\|^2}{\sum_\ell\|A^{(\ell)}\|_F^2}\frac{\|A\|_F^2}{|A(i,j)|^2}\right) &(105)\\
&=\frac{\sum_{\ell=1}^{\tau}|A^{(\ell)}(i,j)|^2}{(\sum_{\ell=1}^{\tau}A^{(\ell)}(i,j))^2}\frac{\|A\|_F^2}{\sum_\ell\|A^{(\ell)}\|_F^2} &(106)\\
&\geq\frac{\|A\|_F^2}{\tau\sum_\ell\|A^{(\ell)}\|_F^2}, &(107)
\end{aligned}$$

where the inequalities follow from the Cauchy-Schwarz inequality. Assumptions 1 and 2 of [FKV04] are satisfied with parameter $c = \frac{\|A\|_F^2}{\tau \sum_\ell \|A^{(\ell)}\|_F^2}$. Hence, this lemma follows as a direction consequence of [FKV04, Lemma 2] with θ substituted by $\theta/(c\tau)$ together with Lemma A.1.

□

Now, we are ready to prove Lemma 5.3.

Lemma 5.3. *Let $A = A^{(1)} + \cdots + A^{(\tau)}$. Let $\epsilon, \eta > 0$ be the error parameter. Take $\mathrm{SQ}(A^{(\ell)})$, $\sigma > 0$, ϵ, and η as the input of Algorithm 4 to obtain $\sigma_1, \ldots, \sigma_r$, $u_1, \ldots, u_r$, and $i_1, \ldots, i_p$. Let $V \in \mathbb{C}^{n\times r}$ be defined as in Eq. (78). Then, with probability at least $9/10$, it holds that $\left\|AVV^\dagger - A\right\|_F^2 \leq \left\|A - A_r\right\|_F^2 + \eta\epsilon^2 \sum_{\ell=1}^{\tau} \left\|A^{(\ell)}\right\|_F$, where A_r is the best rank-r approximation of A. Moreover, r satisfies $\ell_A(\sigma(1+\eta)) \leq r \leq \ell_A(\sigma(1-\eta))$.*

Proof. To begin with, we consider $p = \Theta\big(\frac{(\sum_\ell \|A\|_F^2)^2}{\epsilon^4\sigma^8\eta^2}\big)$. The proof of this lemma most follows the proof of the FVK algorithm. We highlight the differences. First, we use Lemma A.2 instead of [FKV04, Lemma 2]. For [FKV04, Theorem 2], we substitute c with $\frac{\|A\|_F^2}{\tau \sum_\ell \|A^{(\ell)}\|_F^2}$. For [FKV04, Lemma 3], we substitute θ with $\theta/(c\tau)$, where $c = \frac{\|A\|_F^2}{\tau \sum_\ell \|A^{(\ell)}\|_F^2}$. The above substitutions gives the desired dependence on $\sum_\ell \|A^{(\ell)}\|_F^2$ instead of $\|A\|_F^2$. Another difference is that the original FKV algorithm has a threshold $\gamma = O(\epsilon/k$, while we have raised this threshold to $O(1/k)$ effectively. The filter threshold γ was used in the original algorithm because the aim was to find a rank-k approximation. Here, the rank requirement is slightly relaxed, so we do not need a strong threshold. Hence, the number of samples p set in Line 1 in Algorithm 4 suffices. The approximate orthogonality gives the τ^2 dependence. The extra τ dependence comes from a modification of Claim 1 in [FKV04], with the fact that $\|S\|_F^2 \leq \tau \sum_\ell \|S^{(\ell)}\|_F^2 \leq 2\tau \sum_\ell \|A^{(\ell)}\|_F^2$. To obtain the claimed bound, we substitute $\epsilon^2\eta$ with $\epsilon^2\eta/\tau^3$ as in the choice of p in Line 1 of Algorithm 4.

To bound r, first note that this algorithm can compute the first r singular values with cumulative additive error at most $\epsilon^2\eta\sqrt{\sum_\ell \left\|A^{(\ell)}\right\|_F}/\tau$. Thus, with high probability, it holds that the minimum singular value is at least $\sigma - \epsilon^2\eta\sqrt{\sum_\ell \left\|A^{(\ell)}\right\|_F}/\tau = \sigma(1 - \epsilon^2\eta\sqrt{\sum_\ell \left\|A^{(\ell)}\right\|_F}/(\sigma\tau))$. □

Corollary 5.4. *With probability at least $9/10$, it holds that $\left\|AVV^\dagger - A_{\sigma,\eta}\right\|_F \leq \epsilon\sqrt{\sum_\ell \left\|A^{(\ell)}\right\|_F^2}/\sqrt{\eta}$.*

Proof. This follows from [Tan19, Corollary 4.8]. □

Lemma 5.5 (Approximate orthogonality)**.** *Let $A = A^{(1)} + \cdots + A^{(\tau)}$. Let $\epsilon, \eta > 0$ be the error parameter. Take $\mathrm{SQ}(A^{(\ell)})$, $\sigma > 0$, ϵ, and η as the input of Algorithm 4 to obtain $\sigma_1, \ldots, \sigma_r$, $u_1, \ldots, u_r$, and $i_1, \ldots, i_p$. Let $V \in \mathbb{C}^{n\times r}$ be defined as in Eq. (78). Then, with probability at least $9/10$, V is $\mathcal{O}\big(\eta\epsilon^2/\tau\big)$-approximate isometry.*

Proof. Most of the arguments in this proof are similar to the proofs in [Tan19, Lemma 6.6, Corollary 6.7, Proposition 6.11]. Let $v_j \in \mathbb{C}^n$ denote the column vector $V(\cdot, j)$, i.e., $v_j = \frac{S^\dagger}{\hat{\sigma}_j}\hat{u}_j$. Choose $\theta = \tau\sqrt{40/p}$. When $i \neq j$, with probability at least $9/10$, it holds that

$$|v_i^\dagger v_j| = \frac{|\hat{u}_i^\dagger SS^\dagger \hat{u}_j|}{\hat{\sigma}_i\hat{\sigma}_j} \leq \frac{|\hat{u}_i^\dagger (SS^\dagger - WW^\dagger)\hat{u}_j|}{\hat{\sigma}_i\hat{\sigma}_j} \leq \frac{\theta \sum_{\ell=1}^{\tau} \left\|S^{(\ell)}\right\|_F^2}{\sigma^2} \leq \frac{2\theta \sum_{\ell=1}^{\tau} \left\|A^{(\ell)}\right\|_F^2}{\sigma^2}, \tag{108}$$

where the second inequality follows from Lemma A.2, and the last inequality uses Lemma A.1. Similarly, when $i = j$, the following holds with probability at least 9/10.

$$|\|v_i\|^2 - 1| = \frac{|\hat{u}_i^\dagger SS^\dagger \hat{u}_i - \hat{\sigma}_i^2|}{\hat{\sigma}_i^2} \leq \frac{|\hat{u}_i^\dagger (SS^\dagger - WW^\dagger)\hat{u}_i|}{\hat{\sigma}_i^2} \leq \frac{2\theta \sum_{\ell=1}^{\tau} \left\|A^{(\ell)}\right\|_F^2}{\sigma^2}. \tag{109}$$

Since $|(V^\dagger V)(i,j)| = |v_i^\dagger v_j|$, each diagonal entry of $V^\dagger V$ is at most $\frac{2\theta \sum_{\ell=1}^{\tau} \left\|A^{(\ell)}\right\|_F^2}{\sigma^2}$ away from 1 and each off-diagonal entry is at most $\frac{2\theta \sum_{\ell=1}^{\tau} \left\|A^{(\ell)}\right\|_F^2}{\sigma^2}$ away from 0. More precisely, let $M \in \mathbb{C}^{n\times n}$ be the matrix with all ones, i.e., $M(i,j) = 1$ for all $i, j \in \{1, \ldots, n\}$, then for all $i, j \in \{1, \ldots, n\}$, we have

$$\left(I - \frac{2\theta \sum_{\ell=1}^{\tau} \left\|A^{(\ell)}\right\|_F^2}{\sigma^2} M\right)(i,j) \leq (V^\dagger V)(i,j) \leq \left(I + \frac{2\theta \sum_{\ell=1}^{\tau} \left\|A^{(\ell)}\right\|_F^2}{\sigma^2} M\right)(i,j). \tag{110}$$

Now, consider the singular-value decomposition of V as $V = AD_V B^\dagger$ where D_V is a square diagonal matrix. Eq. (110) implies that $\left\|D_V - I\right\|_F \lesssim \frac{2\theta\tau(\sum_{\ell=1}^{\tau} \|A^{(\ell)}\|_F^2)^2}{\sigma^4}$. Take $U := AB^\dagger$. We have $\|U - V\| \leq \left\|U - V\right\|_F \lesssim \eta\epsilon^2/\tau$ by the choice of p in Line 1 in Algorithm 4. Then this lemma follows. $\square$

Lemma 5.7. *Given the succinct description of V, with probability at least 9/10, we can obtain* $\mathrm{SQ}_{\eta\epsilon^2/\tau}^{\eta\epsilon^2/\tau}(V)$ *with* $\mathbf{q}(V) = \mathcal{O}\Big(\frac{\tau^7(\sum_\ell \|A^{(\ell)}\|_F^2)^4}{\epsilon^4\sigma^8\eta^2}\Big)$, $\mathbf{s}(V) = \mathcal{O}\Big(\frac{\tau^{13}(\sum_\ell \|A^{(\ell)}\|_F^2)^9}{\epsilon^8\sigma^{18}\eta^4}\Big)$, $\mathbf{n}_{\eta\epsilon^2/\tau}^{\eta\epsilon^2/\tau}(V) = \widetilde{\mathcal{O}}(1)$.

Proof. For the query access, given $i \in [n]$ and $j \in [r]$,

$$V(i,j) = \frac{S^\dagger(i,\cdot)}{\sigma_j} u_j, \tag{111}$$

where $S^\dagger(i,t) := \sum_{\ell=1}^{\tau} A^{(\ell)}(i_t, i)/\sqrt{pP_{i_t}}$ by the definition of S in Eq. (76). Therefore, $V(i,j)$ can be computed in time $\mathcal{O}(p\tau)$. We can obtain the sampling access by following Proposition 3.7. The time complexity will be $\mathcal{O}\Big(p\mathsf{C}_{S^T, u_j/\sigma_j}(s(S) + pq(v))\Big)$, which is bounded by $\mathcal{O}\Big(p^2 \frac{\tau \sum_{\ell=1}^{\tau} \|A^{(\ell)}\|_F^2}{\sigma^2}\Big)$.

The time complexity comes from the choice of p in Line 1 in Algorithm 4. $\square$

B Finding a CUR decomposition

We will use a result about matrix multiplication proved by Drineas, Kannan, and Mahoney [DKM06]. We state their result in a slightly stronger form, which is actually proved in their paper. For completeness we present a proof following their approach. For the tail bound we use the "independent bounded difference inequality" of McDiarmid [McD89].

Lemma B.1 ([McD89, Lemma (1.2)])**.** *Let $X_1, \ldots, X_c$ be independent random variables, with X_s taking values in a set A_s for each $s \in [c]$. Suppose that f is a real valued measurable function on the product set $\Pi_s A_s$ such that $|f(x) - f(x')| \leq b_s$ whenever the vectors x and x' differ only in the s-th coordinate. Let Y be the random variable $f[X_1, \ldots, X_c]$. Then for any $\gamma > 0$:*

$$P[|Y - E[Y]| \geq \gamma] \leq 2e^{-\frac{2\gamma^2}{\sum b_s^2}}. \tag{112}$$

Now we are ready prove the main result on matrix multiplication.

Lemma B.2 (Matrix multiplication by subsampling [DKM06, Theorem 1])**.** *Suppose we are given $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times p}, c \in \mathbb{Z}^+, \{p_i\}_{i=1}^n$ a probability distribution.*

Consider forming C, R by sampling $i_1, \dots, i_c$ from p, and setting the s-th column of C and row of R to $A(\cdot, i_s)/\sqrt{c}p_{i_s}^A$ and $B(i_s, \cdot)/\sqrt{c}p_{i_s}^B$, respectively, where $p_{i_s}^A$ and $p_{i_s}^B$ are any numbers that multiply to p_{i_s}. Then CR is an unbiased estimator for AB and the following further holds. If, for some positive constant $\beta \le 1$,

$$p_k \ge \frac{\beta\|A(\cdot,k)\|\|B(k,\cdot)\|}{\sum\|A(\cdot,\ell)\|\|B(\ell,\cdot)\|}, \tag{113}$$

the product AB can be estimated by the product of $C \in \mathbb{R}^{m\times c}$ and $R \in \mathbb{R}^{c\times p}$, such that

$$\Pr\left[\|AB - CR\|_F < \frac{\eta}{\sqrt{\beta c}}\underbrace{\sum\|A(\cdot,k)\|\|B(k,\cdot)\|}_{\le\|A\|_F\|B\|_F}\right] > 1-\delta, \tag{114}$$

where $\eta = 1 + \sqrt{(4/\beta)\ln(2/\delta)}$. Further, $\|CR\|_F \le \sum_{k=1}^c \|C(\cdot,k)R(k,\cdot)\| \le \frac{1}{\beta}\sum\|A(\cdot,k)\|\|B(k,\cdot)\|$.

Proof. It is easy to see that

$$\mathbb{E}[CR] = c\cdot\mathbb{E}[C(\cdot,1)R(1,0)] = c\cdot\sum_i p_i\frac{A(\cdot,i)B(i,.)}{cp_i} = AB. \tag{115}$$

Since the indices are selected independently we have

$$\begin{aligned}
\mathbb{E}[\|CR-AB\|_F^2] &= \frac{1}{c}\mathbb{E}[\|c\cdot C(\cdot,1)R(1,\cdot) - AB\|_F^2] &\text{(116)}\\
&\le \frac{1}{c}\sum_{i,j}\mathbb{E}[c^2(CR)_{i,j}^2] &\text{(117)}\\
&= \frac{1}{c}\sum_{i,j}\sum_k p_k\frac{1}{p_k^2}A(i,k)^2B(k,j)^2 &\text{(118)}\\
&= \frac{1}{c}\sum_k\frac{1}{p_k}\|A(\cdot,k)\|^2\|B(k,\cdot)\|^2 &\text{(119)}\\
&\le \frac{1}{c}\sum_k\frac{1}{\beta}\|A(\cdot,k)\|\|B(k,\cdot)\|\sum_\ell\|A(\cdot,\ell)\|\|B(\ell,\cdot) &\text{(120)}\\
&= \frac{1}{\beta c}\left(\sum_k\|A(\cdot,k)\|\|B(k,\cdot)\|\right)^2. &\text{(121)}
\end{aligned}$$

By Jensen's inequality we have

$$\mathbb{E}[\|CR-AB\|_F]^2 \le \mathbb{E}[\|CR-AB\|_F^2], \tag{122}$$

so we get

$$\mathbb{E}[\|CR-AB\|_F] \le \frac{1}{\sqrt{\beta c}}\sum_k\|A(\cdot,k)\|\|B(k,\cdot)\|. \tag{123}$$

Let f be the function $[n]^c \to \mathbb{R}$ defined to be

$$f(i_1, i_2, \dots, i_c) := \left\| AB - \sum_s \frac{1}{cp_{i_s}} A(\cdot, i_s) B(i_s, \cdot) \right\|_F, \tag{124}$$

so that $\mathbb{E}[f] = \mathbb{E}[\|CR - AB\|_F]$. Now suppose that the index sequences $\boldsymbol{i}$ and $\boldsymbol{i}'$ only differ at the s-th position. Then using the triangle inequality it is easy to see

$$|f(\boldsymbol{i}) - f(\boldsymbol{i}')| \leq \frac{1}{c} \left\| \frac{1}{p_{i_s}} A(\cdot, i_s) B(i_s, \cdot) - \frac{1}{p_{i'_s}} A(\cdot, i'_s) B(i'_s, \cdot) \right\|_F \tag{125}$$

$$\leq \frac{2}{c} \max_k \left\| \frac{1}{p_k} A(\cdot, k) B(k, \cdot) \right\|_F \tag{126}$$

$$\leq \frac{2}{\beta c} \sum_k \|A(\cdot, k)\| \|B(k, \cdot)\|. \tag{127}$$

Now we use Lemma B.1 to conclude that

$$P\left[|f - E[f]| \geq \sqrt{\frac{4 \ln(2/\delta)}{\beta}} \frac{1}{\sqrt{\beta c}} \sum_k \|A(\cdot, k)\| \|B(k, \cdot)\| \right] \leq \delta. \tag{128}$$

Therefore

$$P\left[\|CR - AB\|_F \geq \left(1 + \sqrt{\frac{4 \ln(2/\delta)}{\beta}} \right) \frac{1}{\sqrt{\beta c}} \sum_k \|A(\cdot, k)\| \|B(k, \cdot)\| \right] \leq \delta. \qquad \square$$

Finally,

$$\|CR\| \leq \sum_{k=1}^c \|C(\cdot, k) R(k, \cdot)\| = \sum_{k=1}^c \frac{\|A(\cdot, k)\| \|B(k, \cdot)\|}{cp_k} \leq \frac{1}{\beta} \sum \|A(\cdot, k)\| \|B(k, \cdot)\|. \tag{129}$$

Corollary B.3 (Approximating $A^\dagger A$ [FKV04])**.** *Suppose we are given $A \in \mathbb{R}^{m \times n}, \epsilon > 0, \delta, \beta \in (0, 1]$, and $\{p_i\}_{i=1}^n$ a probability distribution, such that*

$$p_i \geq \frac{\beta \|A(i, \cdot)\|^2}{\|A\|_F^2}. \tag{130}$$

Let $r \geq \frac{4\|A\|_F^4 \ln(3/\delta^2)}{\beta^2 \epsilon^2}$. Consider forming R by sampling $i_1, \dots, i_r$ from p, and setting the s-th row of R to $A(i_s, \cdot)/\sqrt{cp_{i_s}}$. Then $R^\dagger R$ is an unbiased estimator for $A^\dagger A$ and

$$\Pr\left[\|A^\dagger A - R^\dagger R\|_F > \epsilon \right] < \delta. \tag{131}$$

Moreover, $\|R\|_F \leq \|A\|_F / \sqrt{\beta}$.

Corollary B.4 (Even singular value transformation)**.** *Let the real function $f \colon \mathbb{R}^+ \to \mathbb{C}$ be such that $f(x)$ has Lipschitz-constant L and $g(x) := (f(x) - f(0))/x$ has Lipschitz constant L', and $|g(x)| \leq B$. Suppose we are given $A \in \mathbb{R}^{m \times n}, \epsilon > 0, \delta, \beta \in (0, 1]$, and $\{p_i\}_{i=1}^n$ a probability distribution, such that*

$$p_i \geq \frac{\beta \|A(i, \cdot)\|^2}{\|A\|_F^2}, \tag{132}$$

and for each i a probability distribution $q^{(i)}$ such that

$$q_j^{(i)} \geq \frac{\gamma |A(i,j)|^2}{\|A(i,\cdot)\|^2}. \tag{133}$$

Let $r \geq \widetilde{\mathcal{O}}\Big(\frac{\|A\|_F^4 L^2 \ln(2/\delta^2)}{\beta^2\epsilon^2}\Big)$ and $c \geq \widetilde{\mathcal{O}}\Big(\frac{\|A\|_F^4 \|A\|^4 L'^2 \ln(2/\delta^2)}{\beta^2\gamma^2\epsilon^2}\Big)$. Form R by sampling $i_1,\ldots,i_c$ from p, and setting the s-th row of R to $A(i_s,\cdot)/\sqrt{cp_{i_s}}$. Then form C by sampling $j_1,\ldots,j_c$ column indices of R by first uniformly sampling a row-index s of R then sampling a column index j_s using the distribution $q^{(i_s)}$, and setting $C(\cdot,s) := R(\cdot,j_s)/\sqrt{\frac{1}{c}\sum_{i=1}^c q_{j_s}^{(i)}}$. Then $R^\dagger g(CC^\dagger)R$ is such that

$$\Pr\left[\left\|R^\dagger g(CC^\dagger)R + f(0)I - f(A^\dagger A)\right\| > \epsilon\right] < \delta. \tag{134}$$

Proof. By Corollary B.3 we have that[xi] $\left\|A^\dagger A - R^\dagger R\right\| \leq \widetilde{\mathcal{O}}\big(\frac{\epsilon}{L}\big)$, and $\|R\|^2 \leq \|A\|^2 + \widetilde{\mathcal{O}}(\epsilon/L) = \widetilde{\mathcal{O}}\Big(\|A\|^2\Big)$ (in the last (in)equality we assumed without loss of generality that $\widetilde{\mathcal{O}}(\epsilon/L) = \widetilde{\mathcal{O}}\Big(\|A\|^2\Big)$). By Lemma 6.1 we get that

$$\left\|(f(A^\dagger A) - f(0)I) - (f(R^\dagger R) - f(0)I)\right\| \leq \epsilon/2. \tag{135}$$

We also have that

$$(f(R^\dagger R) - f(0)I) = R^\dagger g(RR^\dagger)R, \tag{136}$$

Finally, note that

$$\left\|R^\dagger g(RR^\dagger)R - R^\dagger g(CC^\dagger)R\right\| \leq \|R\|^2 \left\|g(RR^\dagger) - g(CC^\dagger)\right\| \leq \widetilde{\mathcal{O}}\Big(\|A\|^2 L' \left\|RR^\dagger - CC^\dagger\right\|\Big) \leq \frac{\epsilon}{2}, \tag{137}$$

where penultimate inequality follows from Lemma 6.1 and the last inequality follows from Corollary B.3. We get Eq. (134) by combining Eqs. (135) to (137). ☐

Lemma B.5. *Consider $X \in \mathbb{C}^{m\times n}, Y \in \mathbb{C}^{n\times p}$. Suppose we know two distributions $\{p_i\}_{i\in[n]}$ and $\{q_i\}_{i\in[n]}$ such that*

$$p_i \geq \beta_1 \frac{\|X(\cdot,i)\|^2}{\|X\|_F^2} \qquad \text{and} \qquad q_i \geq \beta_2 \frac{\|Y(i,\cdot)\|^2}{\|Y\|_F^2}. \tag{138}$$

Then we can formulate a distribution $\{r_i\}_{i\in[n]}$ such that

$$r_i \geq \tilde{\beta} \frac{\|X(\cdot,i)\|\|Y(i,\cdot)\|}{\sum_\ell \|X(\cdot,\ell)\|\|Y(\ell,\cdot)\|}, \tag{139}$$

where $\tilde{\beta} = \sqrt{\beta_1\beta_2}\frac{\sum_\ell \|X(\cdot,\ell)\|\|Y(\ell,\cdot)\|}{\|X\|_F\|Y\|_F}$. Namely, that means that when applying Lemma B.2, we get that

$$\Pr\left[\|AB - CR\|_F \lesssim \sqrt{\frac{\log 1/\delta}{\beta_1\beta_2 c}}\|A\|_F\|B\|_F\right] > 1 - \delta.$$

Moreover, we can find $r_i^A \geq p_i$ and $r_i^B \geq q_i$ such that $r_i^A r_i^B = r_i$. Namely, $\|C\|_F \leq \|A\|_F/\sqrt{\beta_1}, \|R\|_F \leq \|B\|_F/\sqrt{\beta_2}$.

[xi] In this proof we locally redefine the notation $\widetilde{\mathcal{O}}(T) := \mathcal{O}\Big(T \cdot \mathrm{polylog}(\frac{T\|A\|_F}{\epsilon\beta})\Big)$.

Proof. Let $r_i = \frac{1}{2}(p_i + q_i)$; that is, let r be the distribution formed by sampling from $\{p_i\}$ and $\{q_i\}$, each with probability $\frac{1}{2}$. Choose $r_i^A = p_i \frac{p_i+q_i}{2\sqrt{p_i q_i}}$, $r_i^B = q_i \frac{p_i+q_i}{2\sqrt{p_i q_i}}$. Then

$$r_i \geq \beta_1 \frac{\|X(\cdot,i)\|^2}{2\|X\|_F^2} + \beta_2 \frac{\|Y(i,\cdot)\|^2}{2\|Y\|_F^2} = \frac{\beta_1 \|X(\cdot,i)\|^2 \|Y\|_F^2 + \beta_2 \|X\|_F^2 \|Y(i,\cdot)\|^2}{2\|X\|_F^2 \|Y\|_F^2}, \tag{140}$$

and hence

$$\begin{aligned} r_i \frac{\sum_\ell \|X(\cdot,\ell)\|\|Y(\ell,\cdot)\|}{\|X(\cdot,i)\|\|Y(i,\cdot)\|} &\geq \frac{\beta_1 \|X(\cdot,i)\|^2 \|Y\|_F^2 + \beta_2 \|X\|_F^2 \|Y(i,\cdot)\|^2}{2\|X(\cdot,i)\|\|Y\|_F \|X\|_F \|Y(i,\cdot)\|} \frac{\sum_\ell \|X(\cdot,\ell)\|\|Y(\ell,\cdot)\|}{\|X\|_F \|Y\|_F} & (141) \\ &\geq \sqrt{\beta_1 \beta_2} \frac{\sum_\ell \|X(\cdot,\ell)\|\|Y(\ell,\cdot)\|}{\|X\|_F \|Y\|_F} =: \tilde{\beta}, & (142) \end{aligned}$$

by the inequality of arithmetic and geometric means. □

Lemma B.6. *If $M = \sum_{t=1}^{\tau} M^{(t)}$, we can sample $s \in [\tau]$ with probability proportional to $\|M^{(t)}\|_F^2$, and we can sample from $m^{(t)}$ the row norms of $M^{(t)}$, then we can sample from a distribution $\{p_i\}$ such that*

$$p_i \geq \beta \frac{\|M(i,\cdot)\|^2}{\|M\|_F^2} \qquad \text{for} \qquad \beta := \frac{\|M\|_F^2}{\tau \sum_t \|M^{(t)}\|_F^2}. \tag{143}$$

Proof. Let p_i be formed by sampling $s \in [\tau]$, then sampling i according to $m^{(s)}$. Then

$$p_i = \sum_{t=1}^{\tau} \frac{\|M^{(t)}\|_F^2}{\sum_t \|M^{(t)}\|_F^2} \frac{\|M_i^{(t)}\|^2}{\|M^{(t)}\|_F^2} = \frac{\sum_t \|M_i^{(t)}\|^2}{\sum_t \|M^{(t)}\|_F^2} \geq \frac{\|\sum_t M_i^{(t)}\|^2}{\tau \sum_t \|M^{(t)}\|_F^2} = \Big(\frac{\|M\|_F^2}{\tau \sum_t \|M^{(t)}\|_F^2}\Big) \frac{\|M_i\|^2}{\|M\|_F^2}, \tag{144}$$

where the inequality follows from the Cauchy-Schwarz inequality. □

Theorem B.7. *Suppose we have $\mathrm{SQ}(A^{(1)}), \mathrm{SQ}((A^{(1)})^\dagger), \ldots, \mathrm{SQ}(A^{(\tau)}), \mathrm{SQ}((A^{(\tau)})^\dagger) \in \mathbb{C}^{m\times n}$, and let $A = A^{(1)} + \cdots + A^{(\tau)}$. Then we can find an approximate* CUR decomposition *of A: $R \in \mathbb{C}^{r\times n}, C \in \mathbb{C}^{m\times c}$ are (normalized) subsets of rows and columns of A, and $U \in \mathbb{C}^{r\times c}$ is such that*

$$\Pr\Big[\|A - CUR\|_2^2 > \epsilon\tau \sum \|A^{(t)}\|_F^2\Big] < \delta, \tag{145}$$

with $r = \tilde{\Theta}(\frac{1}{\epsilon^4}\log\frac{1}{\delta})$ and $c = \tilde{\Theta}(\frac{1}{\epsilon^6}\log\frac{1}{\delta})$. This can be achieved in $\tilde{O}(\frac{1}{\epsilon^{18}}\log^3\frac{1}{\delta} + \frac{1}{\epsilon^{12}}\log^2\frac{1}{\delta}(\sum_i \mathbf{sq}(A^{(i)})))$ time. Further, we can find $M_1 \in \mathbb{C}^{c\times c}, M_2 \in \mathbb{C}^{c\times r}, M_3 \in \mathbb{C}^{r\times r}$ such that $CUR = (CM_1)M_2(M_3R)$, CM_1 and M_3R are α-approximate isometries, and M_2 is diagonal. This can be achieved in $\tilde{O}((\frac{1}{\alpha^2\epsilon^{24}} + \frac{1}{\alpha^4\epsilon^6})\log^3\frac{1}{\delta} + \frac{1}{\alpha^2\epsilon^{18}}\log^2\frac{1}{\delta}(\sum_i \mathbf{sq}(A^{(i)})))$ time.

Proof. Denote $\mathsf{A} = \sqrt{\tau \sum \|A^{(t)}\|_F^2}$. Since this is the normalization factor[xii] for the distributions we use to sample from A, it will appear often.

[xii] When $\tau = 1$, this is just $\|A\|_F^2$. The τ factor makes sense, since then $\|\sum A^{(t)}\|_F \leq \mathsf{A}$, and splitting up the input matrix can only make A larger and our approximations worse.

We begin by applying Lemma B.6 and Lemma B.5 to approximate $A^\dagger A$ by $R^\dagger R$, by sampling $r := \frac{1}{\epsilon_r^2}\log\frac{1}{\delta}$ rows of A. Consider the eigenvalue thresholding function

$$f_\lambda(x) = \begin{cases} \frac{1}{x} & x \geq \lambda \\ \frac{2}{\lambda^2}(x-\lambda/2) & \lambda/2 \leq x < \lambda \\ 0 & x < \lambda/2. \end{cases} \tag{146}$$

Then, consider the quantity $AR^\dagger f_\lambda(RR^\dagger)R$: note that this is the projection of A onto the row space of R, only smoothly thresholded so that the eigenvectors of $RR^\dagger$ with eigenvalues below λ are sent to zero. (Without the thresholding this would simply be AR^+R, the projection of A onto the row space of R.) This is a good spectral approximation of A:

$$\begin{aligned}
\|A - AR^\dagger f_\lambda(RR^\dagger)R\|_2^2 &= \max_{x:\|x\|\leq 1} \|A(I - R^\dagger f_\lambda(RR^\dagger)R)x\|^2 && (147)\\
&\leq \max_{x:\|x\|\leq 1} \|A(I - R^\dagger (RR^\dagger)_\lambda^+ R)x\|^2 && (148)\\
&= \max_{y:(R^\dagger R)_\lambda y=0, \|y\|\leq 1} \|Ay\|^2 && (149)\\
&\leq \max_{y:\|Ry\|^2\leq\lambda, \|y\|\leq 1} \|Ay\|^2 && (150)\\
&\leq \max_{y:\|Ry\|^2\leq\lambda, \|y\|\leq 1} y^\dagger A^\dagger Ay - y^\dagger R^\dagger Ry + \lambda && (151)\\
&\leq \|A^\dagger A - R^\dagger R\|_2 + \lambda && (152)\\
&\leq \epsilon_R \mathsf{A}^2 + \lambda, && (153)
\end{aligned}$$

where the last inequality applies Corollary B.3.

We can compute approximate singular value decompositions for R by estimating $RR^\dagger$ (call this approximation T); to sample columns of R, sample a row i uniformly at random, then sample an $R^{(t)}$ proportional to $\|A^{(t)}(i,\cdot)\|^2$ (or, equivalently, proportional to $\|R^{(t)}(i,\cdot)\|^2$), then sampling from $R^{(t)}(i,\cdot)$. The probability of sampling a column j is

$$\begin{aligned}
\frac{1}{r}\sum_i \frac{\sum_t \|R^{(t)}(i,j)\|^2}{\sum_t \|R^{(t)}(i,\cdot)\|^2} \geq \frac{1}{r}\sum_i \frac{\|R(i,j)\|^2}{\tau\sum_t \|R^{(t)}(i,\cdot)\|^2} &= \sum_i \frac{\|R(i,j)\|^2}{\tau \sum_t \|A^{(t)}\|_F^2} = \frac{\|R(\cdot,j)\|^2}{\tau\sum_t \|A^{(t)}\|_F^2} && (154)\\
&= \frac{\|R(\cdot,j)\|^2}{\|R\|_F^2}\frac{\|R\|_F^2}{\tau\sum_t \|A^{(t)}\|_F^2}. && (155)
\end{aligned}$$

We do the same to approximate $C^\dagger C$ by a matrix Q. So, by Corollary B.3, with $t := \frac{\log(1/\delta)}{\epsilon_T^2}$ and $q := \frac{\log(1/\delta)}{\epsilon_Q^2}$ many samples, respectively,

$$\|RR^\dagger - T\|_F \leq \epsilon_T \mathsf{A}^2 \qquad \|T\|_F \leq \mathsf{A} \qquad \|C^\dagger C - Q\|_F \leq \epsilon_Q \mathsf{A}^2 \qquad \|Q\|_F \leq \mathsf{A}. \tag{156}$$

We can compute T, so we can compute $f_\lambda(T)$, with the error bound following from Lemma 6.1.

$$\begin{aligned}
\left\|AR^\dagger f_\lambda(RR^\dagger)R - AR^\dagger f_\lambda(T)R\right\|_F &\lesssim \|A\|\epsilon_T \frac{\mathsf{A}^2\|R\|_F^2}{\lambda^2}\log^2\frac{\|R\|^2\|T\|}{\epsilon_T\mathsf{A}^2} && (157)\\
&\leq \epsilon_T\frac{\mathsf{A}^5}{\lambda^2}\log^2\frac{\mathsf{A}}{\epsilon_T}. && (158)
\end{aligned}$$

Now, we approximate $AR^\dagger$ by applying Lemma B.6 to get a distribution for Lemma B.5 and Lemma B.2. Let W be the matrix formed by subsampling columns of R according to the scaling. So, using $\frac{\log(1/\delta)}{\epsilon_C^2}$ samples,

$$\begin{aligned}
\|AR^\dagger f_\lambda(T)R - CW^\dagger f_\lambda(T)R\|_F &\leq \|AR^\dagger - CW^\dagger\|_F \|f_\lambda(T)R\| && (159)\\
&\leq \epsilon_C \sqrt{\frac{\mathsf{A}^2}{\|A\|_F^2}\frac{\mathsf{A}^2}{\|R\|_F^2}}\|A\|_F\|R^\dagger\|_F\|f_\lambda(T)R\| && (160)\\
&\leq \epsilon_C \frac{\mathsf{A}^3}{\lambda}. && (161)
\end{aligned}$$

Further, by rescaling according to p_i^A and p_i^B, we can get that $\|C\|_F \leq \mathsf{A}$ and $\|W^\dagger f_\lambda(T)R\|_F \leq \|f_\lambda(T)\|_F\mathsf{A}^2$. Recall that we computed $Q \approx C^\dagger C$ and $T \approx RR^\dagger$. We can further take the spectral decomposition of these matrices to get $Q = VD_QV^\dagger$ and $T = UD_TU^\dagger$. Let $V_{\geq\xi}$ and $V_{<\xi}$ be the eigenvectors in V with corresponding eigenvalues $\geq \xi$ and $< \xi$, respectively, and correspondingly for $U_{\geq\xi}, U_{<\xi}$. Observe that

$$\begin{aligned}
\|CV_{<\xi}\|^2 = \|V_{<\xi}^\dagger C^\dagger CV_{<\xi}\| \leq \|V_{<\xi}^\dagger(C^\dagger C - Q)V_{<\xi}\| + \|V_{<\xi}^\dagger QV_{<\xi}\| \lesssim \epsilon_Q\mathsf{A}^2 + \xi; && (162)\\
\|D_Q^{-\frac{1}{2}}V_{\geq\xi}^\dagger C^\dagger CV_{\geq\xi}D_Q^{-\frac{1}{2}} - I\| = \|D_Q^{-\frac{1}{2}}V_{\geq\xi}^\dagger(C^\dagger C - Q)V_{\geq\xi}D_Q^{-\frac{1}{2}}\| \leq \epsilon_Q\mathsf{A}^2/\xi. && (163)
\end{aligned}$$

So

$$\begin{aligned}
\|CV_{\geq\xi}V_{\geq\xi}^\dagger W^\dagger f_\lambda(T)R - CW^\dagger f_\lambda(T)R\| &\leq \|CV_{<\xi}\|\|W^\dagger f_\lambda(T)R\|_F && (164)\\
&\leq \sqrt{\epsilon_Q\mathsf{A}^2 + \xi}\|f_\lambda(T)\|\mathsf{A}^2 && (165)\\
&\leq \sqrt{\epsilon_Q\mathsf{A}^2 + \xi}\frac{\mathsf{A}^2}{\lambda}. && (166)
\end{aligned}$$

Hence, if we set $\xi = \epsilon_Q\mathsf{A}^2/\alpha$ and $\lambda = \epsilon_T\mathsf{A}^2/\alpha$, the decomposition

$$CV_{\geq\xi}V_{\geq\xi}W^\dagger f_\lambda(T)R = \left(CV_{\geq\xi}D_Q^{-\frac{1}{2}}\right)\left(D_Q^{\frac{1}{2}}V_{\geq\xi}^\dagger W^\dagger U_{\lambda/2}f_\lambda(D_T)D_T^{\frac{1}{2}}\right)\left(D_T^{-\frac{1}{2}}U_{\lambda/2}^\dagger R\right) \tag{167}$$

has first and last parts as α-approximate isometries, as desired. Finally, compute the singular value decomposition of the middle portion to get some $\tilde{U}D\tilde{V}$, and combine $\tilde{U}$ and $\tilde{V}$ with the left and right matrices, respectively, to get the desired output, since

$$\begin{aligned}
\|A - CV_{\geq\xi}V_{\geq\xi}^\dagger W^\dagger f_\lambda(T)R\| \leq{}& \|A - AR^\dagger f_\lambda(RR^\dagger)R\| + \|AR^\dagger f_\lambda(RR^\dagger)R - AR^\dagger f_\lambda(T)R\|\\
&+ \|AR^\dagger f_\lambda(T)R - CW^\dagger f_\lambda(T)R\| + \|CW^\dagger f_\lambda(T)R - CV_{\geq\xi}V_{\geq\xi}^\dagger W^\dagger f_\lambda(T)R\|, && (168)
\end{aligned}$$

which we can write out as

$$\begin{aligned}
\|A - CMR\| &\lesssim \left(\epsilon_R\mathsf{A}^2 + \frac{\lambda}{2}\right)^{1/2} + \|A\|\epsilon_T\frac{\mathsf{A}^4}{\lambda^2}\log^2\frac{\mathsf{A}^2}{\epsilon_T} + \epsilon_C\frac{\mathsf{A}^3}{\lambda} + \sqrt{\epsilon_Q\mathsf{A}^2 + \xi}\frac{\mathsf{A}^2}{\lambda}, \text{and} && (169)\\
\frac{\|A - CMR\|}{\mathsf{A}} &\lesssim \sqrt{\epsilon_R} + \frac{\sqrt{\lambda}}{\mathsf{A}} + \epsilon_T\frac{\mathsf{A}^4}{\lambda^2}\log^2\frac{\mathsf{A}^2}{\epsilon_T} + \epsilon_C\frac{\mathsf{A}^2}{\lambda} + \sqrt{\epsilon_Q + \frac{\xi}{\mathsf{A}^2}}\frac{\mathsf{A}^2}{\lambda}. && (170)
\end{aligned}$$

By setting parameters carefully, we can make $\|A - CMR\| \lesssim \epsilon\mathsf{A}$. In particular, we can choose $\epsilon_R = \tilde{\Theta}(\epsilon^2), \epsilon_C = \tilde{\Theta}(\epsilon^3), \epsilon_T = \tilde{\Theta}(\min(\epsilon^5, \alpha\epsilon^2)), \epsilon_Q = \tilde{\Theta}(\alpha\epsilon^6), \lambda = \tilde{\Theta}(\epsilon^2\mathsf{A}^2), \xi = \tilde{\Theta}(\epsilon^6\mathsf{A}^2)$. For the complexity analysis, note that the procedure is algorithm 7 as follows. The time complexity is dominated by computing the singular value decompositions and reading in all of the sampled entries, giving the time complexities in the theorem statement. To get just the CUR decomposition without the approximate isometry constraints, just stop running the algorithm after approximating $AR^\dagger$ by $CW^\dagger$. $\square$

Algorithm 7: Procedure of the CUR decomposition.

Input : Set of m-by-n matrices $\{A_1, \dots, A_\tau\}$ with $\mathrm{S}(A_\ell)$ and $\mathrm{Q}(A_\ell)$ for $\ell \in [\tau]$ and $\mathrm{rank}(A_\ell) \le r$; error parameter ϵ.

Sample r rows of A to get $R \in \mathbb{C}^{r \times n}$;

Sample t columns of R to get an $\tilde{R} \in \mathbb{C}^{r \times t} \log \frac{1}{\delta}$ such that $T = \tilde{R}\tilde{R}^\dagger$;

Compute the SVD of $\tilde{R}$ to get $U_{\lambda/2}$ and D_T;

Use sampling power to $R^\dagger$ and A to sample w columns to form $C \in \mathbb{C}^{m \times c}$ and $W \in \mathbb{C}^{r \times c}$;

Sample q rows of C to get an $\tilde{C} \in \mathbb{C}^{q \times c}$ such that $Q := \tilde{C}^\dagger \tilde{C}$;

Compute the SVD of $\tilde{C}$ to learn $V_{\ge\xi}$ and D_Q;

Multiply together $D_Q^{\frac{1}{2}} V_{\ge\xi}^\dagger W^\dagger U_{\lambda/2} f_\lambda(D_T) D_T^{\frac{1}{2}}$, and output it, along with $V_{\ge\xi}, D_Q^{-\frac{1}{2}}, D_T^{-\frac{1}{2}}, U_{\lambda/2}$;