SUBLINEAR TIME QUANTUM ALGORITHM FOR ATTENTION APPROXIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Given the query, key and value matrices $Q, K, V \in \mathbb{R}^{n\times d}$, the attention module is defined as $\mathrm{Att}(Q, K, V) = D^{-1}AV$ where $A = \exp(QK^\top/\sqrt{d})$ with $\exp(\cdot)$ applied entrywise, $D = \mathrm{diag}(A\mathbf{1}_n)$. The attention module is the backbone of modern transformers and large language models, but explicitly forming the softmax matrix $D^{-1}A$ incurs $\Omega(n^2)$, motivating numerous approximation schemes that reduce runtime to $\widetilde{O}(nd)$ via sparsity or low-rank factorization.

We propose a quantum data structure that approximates any row of $\mathrm{Att}(Q, K, V)$ using only row queries to Q, K, V. Our algorithm preprocesses these matrices in $\widetilde{O}\left(\epsilon^{-1}n^{0.5}\left(s_\lambda^{2.5} + s_\lambda^{1.5}d + \alpha^{0.5}d\right)\right)$ time, where ϵ is the target accuracy, s_λ is the λ-statistical dimension of the exponential kernel defined by Q and K, and α measures the row distortion of V. Each row query can be answered in $\widetilde{O}(s_\lambda^2 + s_\lambda d)$ time.

To our knowledge, this is the first quantum data structure that approximates rows of the attention matrix in sublinear time with respect to n. Our approach relies on a quantum Nyström approximation of the exponential kernel, quantum multivariate mean estimation for computing D, and quantum leverage score sampling for the multiplication with V.

1 INTRODUCTION

Transformers (Vaswani et al., 2017) have emerged as one of the most successful machine learning architectures in recent years, revolutionizing fields such as natural language processing (Devlin et al., 2019; Yang et al., 2019; Raffel et al., 2020; Brown et al., 2020; Jiao et al., 2020), computer vision (Carion et al., 2020; Dosovitskiy et al., 2021; Guo et al., 2022), speech recognition (Chorowski et al., 2015; Wang et al., 2021), robotics (Liu et al., 2022), and time series forecasting (Zhou et al., 2021). These models typically operate on sequences of length n, autoregressively predicting the next most likely token to produce an output of length n. In applications like large language models (LLMs), it has been widely observed that increasing the sequence length n significantly enhances generative performance. However, this benefit comes at a substantial computational cost: the core attention module has a quadratic time complexity in n, which severely limits both training and inference scalability.

Formally, let $Q, K, V \in \mathbb{R}^{n\times d}$ denote the query, key, and value embeddings. The attention module is defined as $\mathrm{Att}(Q, K, V) = D^{-1}AV \in \mathbb{R}^{n\times d}$, where $A = \exp(QK^\top/\sqrt{d}) \in \mathbb{R}^{n\times n}$ is computed entrywise, and $D = \mathrm{diag}(A\mathbf{1}_n) \in \mathbb{R}^{n\times n}$. The matrix A is referred to as the *attention matrix*, and $D^{-1}A$ as the *softmax matrix*. Due to the $n \times n$ size of A, much recent research has focused on reducing the quadratic complexity by approximating attention through pattern-based sparse attention (Daras et al., 2020; Kitaev et al., 2020; Roy et al., 2021; Sun et al., 2022; Child et al., 2019; Beltagy et al., 2020; Ainslie et al., 2020; Zaheer et al., 2020), linearizing the kernel through feature mapping (Katharopoulos et al., 2020; Choromanski et al., 2021; Wang et al., 2020; Peng et al., 2021), or various algorithmic and data structure optimizations (Zandieh et al., 2023; Alman & Song, 2023; Han et al., 2024; Kacham et al., 2024; Zandieh et al., 2024; van den Brand et al., 2024; Song et al., 2024; Kannan et al., 2025; Chu et al., 2024; Chen et al., 2025b; Indyk et al., 2025).

The theoretical goal in these efforts is to achieve a runtime that scales nearly linearly with n, allowing some approximation error. This is a natural target, since the input size to the attention module is $n \times d$.

On a classical computer, any algorithm that approximates attention in time $\widetilde{O}(nd)$ is considered optimal. But could this process be accelerated further using a quantum computer?

If our objective is to output the entire $n \times d$ matrix $\mathrm{Att}(Q, K, V)$, then $\Omega(nd)$ time is unavoidable due to output size. However, in many transformer applications — particularly during inference (Pope et al., 2023; Brandon et al., 2024; Adnan et al., 2024; Zhang et al., 2024a; Feng et al., 2025; Liu et al., 2024b; Kumari et al., 2024; Behnam et al., 2025; Chen et al., 2025a;c; Indyk et al., 2025) — only *row queries* are needed. In this setting, we aim to preprocess Q, K, V into a data structure such that, for any index $i \in [n]$, the structure can return a vector $\widetilde{r}_i \in \mathbb{R}^d$ that approximates the i-th row of $\mathrm{Att}(Q, K, V)$. This model circumvents the $\Omega(nd)$ lower bound by focusing on partial output. Nonetheless, since each row of $\mathrm{Att}(Q, K, V)$ is a convex combination of rows of V, achieving truly sublinear time in n still appears classically intractable.

In this work, we answer this question affirmatively. Specifically, we construct a quantum data structure that preprocesses Q, K, V using only *row queries*, and does so in time[1] $\widetilde{O}(\epsilon^{-1} n^{0.5} \cdot \mathrm{poly}(d, s_\lambda, \alpha))$, where s_λ is the *statistical dimension* of the exponential kernel matrix associated with Q and K, and α is a measure of the row distortion of V (see Definition C.2). Given any index $i \in [n]$, the data structure returns an approximation to the i-th row of $\mathrm{Att}(Q, K, V)$ in time $\widetilde{O}(s_\lambda^2 + s_\lambda d)$.

To our knowledge, this is the first quantum algorithm to implement the row query model in sublinear time. Prior works either require superlinear preprocessing time or impose structural assumptions (Gao et al., 2023). Our approach avoids both: it makes *no assumptions* on Q, K, V, making it broadly applicable in practice. Moreover, our construction is conceptually simple — it combines quantum techniques such as Grover search (Grover, 1996), Nyström kernel approximation, and quantum multivariate mean estimation (Cornelissen et al., 2022) to approximate each component of the attention module: D, A, and V.

Quantum Computation Model. We follow the standard quantum computation framework as in Apers & De Wolf (2022); Apers & Gribling (2023). The model allows quantum subroutines using $O(\log n)$ qubits, quantum queries to the input, and access to a quantum-read/classical-write RAM (QRAM) of $\mathrm{poly}(n)$ bits. Each quantum read or classical write takes unit cost. We measure *time complexity* by the number of QRAM operations, and *query complexity* by the number of queries to the input. In our setting, we query rows of Q, K, and V, each requiring $O(d)$ time classically. For simplicity, we assume Q and K have been scaled by $1/d^{1/4}$, which can also be done via row queries in $O(d)$ time.

2 Preliminary

Notation. Given symmetric matrices $A, B \in \mathbb{R}^{n \times n}$, we use $A - B \succeq 0$ to denote $A - B$ is a positive semidefinite (PSD) matrix, i.e., for any $x \in \mathbb{R}^n$, $x^\top (A - B) x \geq 0$. Given a matrix $M \in \mathbb{R}^{n \times n}$, we use $\exp(M)$ to denote the entrywise exponentiation operation. We use $\mathrm{tr}[M]$ to denote the trace of M. For a real matrix A, we use $A^\dagger$ to denote its Moore-Penrose pseudoinverse, and for a square, nonsingular real matrix M, we use M^{-1} to denote its inverse. For two vectors $x, y \in \mathbb{R}^n$, we use $x^\top y$ or $\langle x, y \rangle$ to denote the inner product of x and y. We use $\mathbf{0}_n$ and $\mathbf{1}_n$ to denote all-0's and all-1's vector. For a vector $x \in \mathbb{R}^n$, we use $\|x\|_2 = \sqrt{x^\top x}$ to denote its ℓ_2 norm, $\|x\|_\infty = \max_{i \in [n]} |x_i|$ to denote its ℓ_∞ norm. If M is a PSD matrix, then we use $\|x\|_M = \sqrt{x^\top M x}$ to denote the M-energy norm of x. For a matrix A, we use $\|A\|$ to denote its spectral norm and $\|A\|_\infty$ to denote its max row ℓ_1 norm, and $\|A\|_F$ to denote its Frobenius norm. Throughout the paper, we will also exclusively work with weighted sampling matrices, usually denoted by $S \in \mathbb{R}^{n \times s}$ for where s is the total number of samples taken, let $i(j)$ be the index of the i-th sample, then the i-th column of S is $\frac{1}{\sqrt{p_j}} e_j$, where p_j is the probability of choosing the index j. We use $\mathbb{E}[X]$ to denote the expectation of a random variable X. We use $\mathbb{I}[E]$ to denote the indicator of whether event E happens.

Numerical Linear Algebra. We rely on several primitives from numerical linear algebra for fast approximations and provable guarantees.

[1] We use $\widetilde{O}(\cdot)$ to suppress polylogarithmic factors in n, d, s_λ, and $1/\epsilon$.

Definition 2.1 (Leverage score). *Let $A \in \mathbb{R}^{n \times d}$. The i-th leverage score of A is defined as*

$$\tau_i := a_i^\top (A^\top A)^{-1} a_i,$$

where a_i is the i-th row of A.

We will also work exclusively with *kernel matrices*. Given a dataset $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$, we define the exponential kernel matrix $E \in \mathbb{R}^{n \times n}$ by $E_{i,j} = \exp(\langle x_i, x_j \rangle)$. Although E is generally full-rank, our algorithm depends only on a parameter called the λ-*statistical dimension* of E, which may be much smaller than n.

Definition 2.2 (Statistical dimension (Zhang, 2005; Hastie et al., 2009)). *Let $E \in \mathbb{R}^{n \times n}$ be a PSD matrix, and let $\lambda > 0$. The λ-statistical dimension of E is defined as $s_\lambda(E) := \mathrm{tr}[E(E + \lambda I)^{-1}]$. When E is clear from context, we write s_λ for simplicity.*

Note that s_λ is a monotonically decreasing function of λ, and is closely related to the notion of ridge leverage scores.

Definition 2.3 (Ridge leverage score (Alaoui & Mahoney, 2015)). *Let $E \in \mathbb{R}^{n \times n}$ be a kernel matrix and let $\lambda > 0$. The λ-ridge leverage score of the data point x_i is defined as*

$$\tau_i^\lambda := (E(E + \lambda I)^{-1})_{i,i}.$$

If $E = BB^\top$ for some $B \in \mathbb{R}^{n \times n}$, then this can be equivalently written as

$$\tau_i^\lambda = b_i^\top (B^\top B + \lambda I)^{-1} b_i,$$

where b_i is the i-th row of B.

It is easy to see that $\sum_{i=1}^n \tau_i^\lambda = s_\lambda$. Moreover, Musco & Musco (2017) shows that Nyström approximations (Williams & Seeger, 2000) based on ridge leverage score sampling yield accurate spectral approximations to E.

Lemma 2.4 (Theorem 3 of Musco & Musco (2017)). *Let $s = O(s_\lambda \log(s_\lambda/\delta))$, $\lambda > 0$, and $\delta \in (0, 1)$. Let $E \in \mathbb{R}^{n \times n}$ be any kernel matrix. Let $S \in \mathbb{R}^{n \times s}$ be the λ-ridge leverage score sampling matrix. Then the Nyström approximation $\widetilde{E} := ES(S^\top ES)^\dagger S^\top E$ satisfies $E \preceq \widetilde{E} \preceq E + \lambda I$ with probability at least $1 - \delta$.*

Quantum Primitives. In this paper, we primarily leverage two quantum algorithmic primitives. The first is an efficient quantum sampling oracle based on Grover search.

Lemma 2.5 (Claim 3 in Apers & De Wolf (2022)). *Let n be a positive integer, and let $\{p_1, \ldots, p_n\} \subseteq [0, 1]$ be a list of probabilities. There exists a quantum algorithm, $\mathrm{QS{\small AMPLE}}(p)$, that generates a list of indices where each i is sampled independently with probability p_i, in time $\widetilde{O}\left(\sqrt{n \sum_{i=1}^n p_i}\right) \cdot \mathcal{T}$, where $\mathcal{T}$ denotes the time required to generate any individual p_i.*

The second primitive is a quantum procedure for approximating matrix-vector products using quantum multivariate mean estimation.

Lemma 2.6 (Theorem 5.1 of Apers & Gribling (2023)). *Let $\epsilon \in (0, 1)$, and let $A \in \mathbb{R}^{n \times d}$ and $v \in \mathbb{R}^n$. Suppose we are given quantum query access to the rows of A and the entries of v. Then there exists a quantum algorithm $\mathrm{QM{\small AT}V{\small EC}}(A, v, \epsilon)$ that outputs a vector $\widetilde{\mu} \in \mathbb{R}^d$ such that, with probability at least $1 - 1/\mathrm{poly}(n)$, $\|\widetilde{\mu} - A^\top v\|_{(A^\top A)^{-1}} \leq \epsilon$, using $\widetilde{O}\left(\epsilon^{-1} n^{0.5} d^{0.5} \|v\|_\infty\right)$ queries to A and v.*

3 Technical Overview

In this section, we provide an overview on the algorithmic techniques we utilize to approximate A, D and V, in sublinear time.

3.1 Approximate the Attention Matrix via Quantum Nyström

To approximate the attention matrix A, we will make use of Nyström approximation (Williams & Seeger, 2000). However, recall that $A = \exp(QK^\top)$; for $Q \neq K$, the matrix itself is not even symmetric. This poses significant challenges for obtaining a good approximation. On the other hand, if we treat the queries and keys as the *dataset*, and form the exponential kernel matrix over them, then the resulting matrix is indeed a kernel matrix.

Specifically, let the dataset $X = \{q_1, \ldots, q_n, k_1, \ldots, k_n\}$, and consider $E \in \mathbb{R}^{2n\times 2n}$ where $E = \begin{bmatrix} \exp(QQ^\top) & \exp(QK^\top) \\ \exp(KQ^\top) & \exp(KK^\top) \end{bmatrix}$, then the attention matrix can be retrieved via $PE\begin{bmatrix} \mathbf{0}_n \\ \mathbf{1}_n \end{bmatrix}$ where $P \in \mathbb{R}^{n\times 2n}$ is the matrix consisting of the first n rows of the $2n \times 2n$ identity matrix, which selects the first n rows of E. Thus, once we obtain an approximation for E, we automatically obtain an approximation for A.

It remains to compute a Nyström approximation of E, as at first glance it is not clear how to even generate the ridge leverage score sampling matrix S in sublinear time. Musco & Musco (2017) shows that on a classical computer, it is possible to compute a *generalized* ridge leverage score sampling matrix using $\widetilde{O}(ns_\lambda)$ evaluations of the kernel function and an additional $\widetilde{O}(ns_\lambda^2)$ time, via a recursive sampling scheme:

- Uniformly sample half of the data points, then recursively compute the weighted sampling matrix $\widetilde{S}^{n\times s}$ for the subset;
- Compute the *generalized ridge leverage score*, defined as $\widetilde{\tau}_i^\lambda := b_i^\top (B^\top \widetilde{S}\widetilde{S}^\top B + \lambda I)^\dagger b_i$, and set $p_i = \min\{1, \widetilde{\tau}_i^\lambda \cdot \log(s_\lambda/\delta)\}$;
- Output S as the weighted sampling matrix according to p_i.

The key ingredients in their algorithm are (1) the generalized ridge leverage score can be computed via kernel function evaluations instead of computing the factorization (see Definition A.4), and (2) sampling according to generalized ridge leverage score only increases the sample size by a constant factor, hence it does not affect the asymptotic runtime of the algorithm (see Lemma A.3).

For the simpler setting of leverage score sampling, Apers & Gribling (2023) shows that this recursive framework can benefit from quantum speedup, especially the Grover search sampler of Lemma 2.5, by noting that when sampling according to the leverage score, it is not necessary to compute or approximate all the scores; rather, it is enough to implement an oracle that can supply any approximate leverage score when needed.

For our application, however, this oracle is much more difficult to implement, as in the setting of Apers & Gribling (2023), one could directly query the row of B, which is not the case for the kernel setting. Nevertheless, we show how to implement such an oracle for generalized ridge leverage scores of kernels. The algorithm is detailed in Algorithm 1. Throughout this section, we let s denote the final sample size of the Nyström approximation.

The main idea is to utilize the identity $\widetilde{\tau}_i^\lambda = \frac{1}{\lambda}(E - ES(S^\top ES + \lambda I)^{-1}S^\top E)_{i,i}$, where $E_{i,i}$ involves a single kernel evaluation $\mathsf{K}(x_i, x_i)$, and $S^\top ES$ requires only $O(s^2)$ kernel evaluations. Finally, the term $(ES(S^\top ES + \lambda I)^\dagger S^\top E)_{i,i}$ can be computed by evaluating the kernel between x_i and the sampled points in S, weighted appropriately, which requires $O(s)$ kernel evaluations. This shows that we can implement the oracle by precomputing $(S^\top ES + \lambda I)^\dagger$ in $O(s^2) \cdot \mathcal{T}_{\mathsf{K}} + s^\omega$ time, where $\mathcal{T}_{\mathsf{K}}$ denotes the time for kernel evaluation and $\omega \approx 2.37$ is the matrix multiplication exponent (Duan et al., 2023; Williams et al., 2024; Alman et al., 2025). Each oracle query can then be answered in $O(s) \cdot \mathcal{T}_{\mathsf{K}} + s^2$ time. By Lemma 2.5, the quantum sampler requires only $\widetilde{O}(n^{0.5}s^{0.5})$ oracle calls, so the overall runtime is $\widetilde{O}(n^{0.5}s^{1.5} \cdot (\mathcal{T}_{\mathsf{K}} + s) + s^\omega)$. In our setting, the kernel function $\mathsf{K}(x_i, x_j) = \exp(\langle x_i, x_j\rangle)$ can be computed in $O(d)$ time, which gives a runtime of $\widetilde{O}(n^{0.5}s^{1.5}(d + s) + s^\omega)$, sublinear in n.

It remains to analyze the approximation guarantee. Sampling according to generalized ridge leverage scores ensures that $E \preceq \widetilde{E} \preceq E + \lambda I$, but this does not immediately imply a bound on the approximation error for $\exp(QK^\top)$. To address this, let $E = \begin{bmatrix} B & A \\ A^\top & C \end{bmatrix}$ and $\widetilde{E} = \begin{bmatrix} \widetilde{B} & \widetilde{A} \\ \widetilde{A}^\top & \widetilde{C} \end{bmatrix}$.

Algorithm 1 Quantum Nyström approximation via recursive generalized ridge leverage score sampling.

procedure $\textsc{QNyströmKernel}(\{x_1, \ldots, x_n\} \in (\mathbb{R}^d)^n, \mathsf{K} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^m, \delta \in (0,1), \lambda \in (0, \infty))$ $\triangleright$ δ is the failure probability, λ is the ridge leverage score parameter.
 $s \leftarrow O(s_\lambda \log(s_\lambda/\delta))$
 $T \leftarrow O(\log(n/s))$
 Let $S_0 \subset_{1/2} S_1 \subset_{1/2} \cdots \subset_{1/2} S_T = [n]$ $\triangleright$ We use $A \subset_{1/2} B$ to denote A is a uniform subset of half of the indices of B
 $M_0 \leftarrow \{\mathsf{K}(x_i, x_j)\}_{(i,j)\in S_0 \times S_0}$ $\triangleright$ $|S_0| = s$
 Let $D_0 \in \mathbb{R}^{n\times|S_0|}$ be the sampling matrix of S_0
 for $t = 1$ to T **do**
  $\widehat{M} \leftarrow (M_{t-1} + \lambda I_s)^{-1}$
  $\triangleright$ Let $D_{t-1}^\top K_i := \{D_{t-1}(j) \cdot \mathsf{K}(x_i, x_j)\}_{j\in D_{t-1}} \in \mathbb{R}^s$ for $i \in S_t$ where $D_{t-1}(j)$ is the weight corresponding to x_j specified by D_{t-1}
  Implement oracle for $q_i \leftarrow \frac{5}{\lambda} \cdot (\mathsf{K}(x_i, x_i) - (D_{t-1}^\top K_i)^\top \widehat{M} D_{t-1}^\top K_i)$ for $i \in S_t$
  $\triangleright$ $p_i = \min\{1, 16 q_i \log(2s/\delta)\}$
  $\widetilde{D}_t \leftarrow \textsc{QSample}(p)$ $\triangleright$ $\widetilde{D}_t \in \mathbb{R}^{|S_t|\times s}$
  $D_t \leftarrow D_{S_t} \cdot \widetilde{D}_t$ $\triangleright$ $D_t \in \mathbb{R}^{n\times s}$
  $M_t \leftarrow \{D_t(i) D_t(j) \cdot \mathsf{K}(x_i, x_j)\}_{(i,j)\in D_t\times D_t}$ $\triangleright$ $M_t \in \mathbb{R}^{s\times s}$
 end for
 return D_T
end procedure

Standard spectral approximation theory guarantees that $B \preceq \widetilde{B} \preceq B + \lambda I$ and $C \preceq \widetilde{C} \preceq C + \lambda I$. By considering all vectors of the form $\begin{bmatrix} x \\ x \end{bmatrix}$, one can show that a similar bound holds for the *symmetrization* of $\exp(QK^\top)$: $A + A^\top \preceq \widetilde{A} + \widetilde{A}^\top \preceq A + A^\top + 2\lambda I$. While this does not directly bound $\widetilde{A}$ in terms of A, it is consistent with our approximation framework, which preserves only the symmetric part of A. Hence, our guarantee holds for the symmetrized attention matrix. It is also worth noting that Algorithm 1 merely computes the weighted sampling matrix S, which can be stored compactly by recording the sampled indices and corresponding weights, but does not explicitly form the Nyström approximation $\widetilde{E} = ES(S^\top ES)^\dagger S^\top E$. While $(S^\top ES)^\dagger$ can be computed and stored in $O(s^2 d + s^\omega)$ time, forming $\widetilde{E}$ would take $\Omega(ns)$ time, which is prohibitive due to output size. In what follows, we show that this restricted representation of S is nonetheless sufficient to approximate D, V, and $\mathrm{Att}(Q, K, V)$.

We now compare our Nyström approximation scheme to a related method known as Nyströmformer (Xiong et al., 2021), which also integrates Nyström into the attention mechanism. Specifically, they consider the attention matrix A and partition it as $A = \begin{bmatrix} X_1 & X_2 \\ X_3 & X_4 \end{bmatrix}$, aiming to approximate X_4 using the other three blocks. Given Nyström landmark points Q' and K' sampled from Q and K, they set $X_1 = \exp(Q'K'^\top)$, $X_2 = \exp(QK'^\top)$, and $X_3 = \exp(Q'K^\top)$. Since the number of landmarks is small, these blocks are all low-dimensional. Xiong et al. (2021) proves that X_4 can be efficiently approximated using X_1, X_2, and X_3 in $O(nmd)$ time, where m is the number of landmarks. While Nyströmformer performs well in practice, it guarantees convergence to the true attention matrix only when all rows of Q and K are included as landmarks. In contrast, our Nyström scheme operates on the exponential kernel matrix formed from Q and K, and achieves spectral approximation guarantees as long as the sample size is sufficiently large without needing to include all data points.

3.2 Approximate the Normalization Factor via Quantum Mean Estimation

Recall that $D = \mathrm{diag}(A\mathbf{1}_n)$, and each normalization factor only requires computing $a_i^\top \mathbf{1}_n$, where a_i is the i-th row of A. If we have access to $\widetilde{E}$, then the i-th normalization factor could be estimated as $\widetilde{E}_{i,*}^\top \begin{bmatrix} \mathbf{0}_n \\ \mathbf{1}_n \end{bmatrix}$. However, as discussed earlier, we cannot explicitly form $\widetilde{E}$ due to its size. To resolve this, we define $U := ES(S^\top ES)^{\dagger/2} \in \mathbb{R}^{2n\times s}$. By the definition of the Nyström approximation, we have $\widetilde{E} = UU^\top$. Given any vector $v \in \mathbb{R}^{2n}$, if we can compute or approximate $U^\top v$, then the normalization factor for the i-th row can be estimated as $u_i^\top(U^\top v)$, where $u_i \in \mathbb{R}^s$ is the i-th row of U. Fortunately, we can implement row queries to U. We first precompute $(S^\top ES)^{\dagger/2}$ in $O(s^2 d + s^\omega)$ time, then each row u_i of U is computed via kernel evaluations between x_i and the points in S, followed by matrix-vector multiplication with $(S^\top ES)^{\dagger/2}$. This takes $O(s^2 + sd)$ time.

It remains to approximate $U^\top v$, which we cast as a multivariate mean estimation problem. Define the random variable $X = 2nv_i U_{*,i}$, where $i \in [2n]$ is selected uniformly at random. It is easy to verify that $\mathbb{E}[X] = U^\top v$, and the variance is bounded. Therefore, one can apply the quantum multivariate mean estimation procedure of Cornelissen et al. (2022) to approximate $U^\top v$. To further reduce variance, Apers & Gribling (2023) proposes approximating the matrix-vector product in the $(U^\top U)^{-1}$-energy norm. Following this idea, we apply Lemma 2.6 to output a vector $\widetilde{\mu} \in \mathbb{R}^s$ such that $\|\widetilde{\mu} - U^\top v\|_{(U^\top U)^{-1}} \le \epsilon$, using $\widetilde{O}(\epsilon^{-1} n^{0.5} s^{0.5} \|v\|_\infty)$ row queries to U and v. In our application, we always have $\|v\|_\infty = 1$, and as noted above, each row query to U takes $O(s^2 + sd)$ time. We present the full algorithm below in Algorithm 2.

Algorithm 2 Algorithm for estimating normalization factor.

data structure QRowNorm
begin members
  $s \in \mathbb{N}$
  $S \in (\mathbb{R}^2)^s$
  $N \in \mathbb{R}^{s\times s}$
  $\widetilde{\mu} \in \mathbb{R}^s$
end members

procedure Preprocess($Q \in \mathbb{R}^{n\times d}, K \in \mathbb{R}^{n\times d}, \lambda \in (0,\infty), \epsilon \in (0,1)$)
  $s \leftarrow O(s_\lambda \log(s_\lambda n))$
  $S \leftarrow$ QNyströmKernel($Q \cup K, (x_i, x_j) \mapsto \exp(\langle x_i, x_j\rangle), 1/\operatorname{poly}(n), \lambda$) ▷ Algorithm 1, S is a list of sampled indices and weights
  $N \leftarrow (S^\top ES)^{\dagger/2}$
  Implement row oracle u_j as follows:
    $\widetilde{u}_{j(k)} \leftarrow S_k \cdot \exp(\langle x_j, x_k\rangle), \forall k \in S$ ▷ $\widetilde{u}_j \in \mathbb{R}^s$
    $u_j \leftarrow N\widetilde{u}_j$ ▷ S stores pairs of indices and weights, S_k is the weight corresponding to index k, $u_j \in \mathbb{R}^s$
  Implement entry oracle for a vector $v \in \mathbb{R}^{2n}$, where $v_j = 0$ for $j \in [n]$ and $v_j = 1$ for $j \in \{n+1, \dots, 2n\}$ ▷ $v = [\mathbf{0}_n; \mathbf{1}_n]^\top$
  $\widetilde{\mu} \leftarrow$ QMatVec(U, v, ϵ) ▷ $\widetilde{\mu} \in \mathbb{R}^s$, Lemma 2.6
end procedure

procedure Query($i \in [n]$)
  $b_i \leftarrow \langle u_i, \widetilde{\mu}\rangle$ ▷ u_i is computed via row oracle
  return b_i
end procedure
end data structure

For the approximation guarantee, we prove that for any vector $x \in \mathbb{R}^s$, if we have $\|x\|_{(U^\top U)^{-1}} \le \epsilon$, then $\|x\|_{U^\top U} \le \epsilon \cdot \|U^\top U\|$. This is particularly useful for us, as we can set $x = U^\top v - \widetilde{\mu}$, in which case $\|x\|_{U^\top U} = \sqrt{x^\top U^\top U x} = \|UU^\top v - U\widetilde{\mu}\|_2$, and the upper bound becomes $\epsilon \cdot \|U^\top U\| = \epsilon \cdot \|\widetilde{E}\| \le \epsilon \cdot (\lambda + \|E\|)$. On the other hand, we can upper bound $\|(\widetilde{E} - E)v\|_\infty$ using the matrix infinity norm, defined as $\|\widetilde{E} - E\|_\infty = \max_{i\in[2n]} \|\widetilde{E}_{i,*} - E_{i,*}\|_1$. A simple argument shows that $\|\widetilde{E} - E\|_\infty \le \sqrt{n} \cdot \|\widetilde{E} - E\| \le \lambda\sqrt{n}$. A triangle inequality then yields the final approximation guarantee. If we define $\widetilde{D} := \operatorname{diag}(\widetilde{A}\mathbf{1}_n)$, the above analysis provides a bound on $\|D - \widetilde{D}\|$. However, in forming the attention module, it is more desirable to control $\|\widetilde{D}^{-1}\|$. To achieve this, we prove a perturbation bound on matrix inversion that relates $\|\widetilde{D}^{-1}\|$ to $\|D^{-1}\|$.

3.3 Approximate the Value Matrix via Leverage Score Sampling

In preceding discussions, we have shown how to construct the sampling matrix for Nyström approximation and how to compute the normalization factor for any row $i \in [n]$. It remains to approximate V in sublinear time. Prior classical algorithms, such as Zandieh et al. (2023), propose using importance sampling based on the *joint row norm* of V and $D^{-1}A$. Specifically, the sampling probability for the i-th row is set as $p_i \geq 1/4 \cdot (\|e_i^\top D^{-1}A\|_2^2 + \gamma \cdot \|v_i\|_2^2)/(\|D^{-1}A\|_F^2 + \gamma \cdot \|V\|_F^2)$, where $\gamma = \|D^{-1}A\|^2/\|V\|^2$. This method achieves a final sample size that is nearly linear in $d + \mathrm{srank}(D^{-1}A)$, where $\mathrm{srank}(D^{-1}A) = \|D^{-1}A\|_F^2/\|D^{-1}A\|^2$ is the stable rank of the softmax matrix. While this approach is conceptually simple and easy to implement, it requires estimating the Frobenius norms of both V and $D^{-1}A$ to constant-factor accuracy. This is straightforward if we are allowed to read all entries of V, but becomes particularly challenging in sublinear time. Our solution is to instead use leverage score sampling on the matrix V, which can be implemented in sublinear time Apers & Gribling (2023). Unlike the joint sampling distribution of Zandieh et al. (2023), which yields a *spectral norm approximate matrix multiplication* guarantee of the form $\|D^{-1}ASS^\top V\| \leq \epsilon \cdot \|D^{-1}A\| \cdot \|V\|$, leverage score sampling has two key limitations: (1) it requires that V have orthonormal columns (Clarkson & Woodruff, 2017), and (2) it provides approximate matrix multiplication guarantees in Frobenius norm, i.e., $\|D^{-1}ASS^\top V\|_F \leq \epsilon \cdot \|D^{-1}A\|_F \cdot \|V\|_F$.

To address the first limitation, we introduce a new parameter called the *row distortion* of V, defined as $\alpha := d/\|V\|_F^2 \cdot \max_{i \in [n]} \|v_i\|_2^2/\tau_i$. Intuitively, α measures the mismatch between the row density and row importance. Specifically, the ratio $\|v_i\|_2^2/\|V\|_F^2$ quantifies how much row v_i contributes in ℓ_2^2 norm, while τ_i/d measures how linearly independent v_i is compared to other rows via τ_i.

Our main result is that by sampling $\widetilde{O}(\epsilon^{-2}\alpha)$ rows of V according to its leverage score distribution, we obtain an approximate matrix multiplication guarantee in Frobenius norm. Note that $\alpha = 1$ if V has orthonormal columns, which recovers the result of Clarkson & Woodruff (2017). This sampling procedure can be implemented in $\widetilde{O}(\epsilon^{-1}n^{0.5}\alpha^{0.5}d)$ time by making row queries to V.

3.4 Main Result

Now that we have described how to approximate each of the matrices D, A, and V, we are in a position to state our main result. We provide an overview of our algorithm below in Algorithm 3.

Theorem 3.1 (Informal version of Theorem D.2)**.** *Let $Q, K, V \in \mathbb{R}^{n \times d}$ be the query, key and value matrices, let $\epsilon, \lambda > 0$. Let $E \in \mathbb{R}^{2n \times 2n}$ be the exponential kernel matrix on the dataset $Q \cup K$ and s_λ be the statistical dimension of E (Definition 2.2) and α be the row distortion of V (Definition C.2). Assume that $\|D^{-1}\| < \frac{1}{\epsilon\|E\| + \lambda\sqrt{n}}$ and let $\beta = \frac{1}{1 - (\epsilon\|E\| + \lambda\sqrt{n})\|D^{-1}\|}$. There exists a quantum data structure that preprocesses Q, K, V through only row queries to these matrices and maintains matrices $\widetilde{D}, \widetilde{A}, \widetilde{V}$ implicitly such that, with probability at least $1 - 1/\mathrm{poly}(n)$,*

$$\|\widetilde{D}^{-1}(\widetilde{A} + \widetilde{A}^\top)/2 \cdot \widetilde{V} - D^{-1}(A + A^\top)/2 \cdot V\|_F$$
$$\leq \epsilon \cdot (\beta \cdot \|D^{-1}\|) \cdot (\|(A + A^\top)/2)\|_F + \lambda\sqrt{n}) \cdot \|V\|_F.$$

Moreover, the data structure has the specification

- *It preprocesses Q, K, V in $\widetilde{O}(\epsilon^{-1}n^{0.5}(s_\lambda^{2.5} + s_\lambda^{1.5}d + \alpha^{0.5}d))$ time;*
- *For any $i \in [n]$, it returns a vector $\widetilde{r}_i = e_i^\top \widetilde{D}^{-1}\widetilde{A}\widetilde{V}$ in $\widetilde{O}(s_\lambda^2 + s_\lambda d)$ time.*

We pause to make some remarks on Theorem 3.1. The preprocessing time scales as $n^{0.5}$, achieving a quadratic speedup over any classical algorithm. Several parameters merit further discussion, in particular the statistical dimension s_λ and the approximation factor for $\|D^{-1}\|$, denoted by β. We summarize their relationships as functions of λ in Table 1. The row distortion factor α also affects the runtime, and the algorithm remains sublinear in n only when $\alpha = o(n)$. As previously illustrated, we have $\alpha = 1$ when V has orthonormal columns. If all rows of V are identical, then $\tau_i = d/n$ and $\|v_i\|_2^2/\|V\|_F^2 = 1/n$ for all $i \in [n]$, leading again to $\alpha = 1$. The distortion factor becomes large only when there exists a row with small leverage score but disproportionately large row norm compared to others. For most practical datasets, one can expect $\alpha = O(1)$.

We also highlight the slightly unusual approximation guarantee: instead of bounding $\|\widetilde{D}^{-1}\widetilde{A}\widetilde{V} - \mathrm{Att}(Q, K, V)\|_F$, we must consider a symmetrization of $\widetilde{A}$. Unfortunately, this is unavoidable given

Algorithm 3 Quantum data structure for attention row query.

data structure QAttention ▷ Theorem 3.1
begin members
  $s_E, s_V \in \mathbb{N}$
  $\widetilde{V} \in \mathbb{R}^{s_V \times d}$
  $\widetilde{N} \in \mathbb{R}^{s_E \times s_V}$
  $\widetilde{L} \in \mathbb{R}^{s_E \times d}$
  QRowNorm QRN ▷ Algorithm 2
end members

procedure Preprocess($Q \in \mathbb{R}^{n \times d}, K \in \mathbb{R}^{n \times d}, V \in \mathbb{R}^{n \times d}, \lambda > 0, \epsilon > 0, \alpha \geq 1$)
  $s_\lambda \leftarrow s_\lambda(E)$
  $s_V \leftarrow \widetilde{O}(\epsilon^{-2}\alpha), s_E \leftarrow \widetilde{O}(s_\lambda)$
  QRN.Preprocess(Q, K, λ, ϵ) ▷ Algorithm 2
  $S_V \leftarrow$ QLeverageScore(V, s_V) ▷ $S_V \in \mathbb{R}^{n \times s_V}$, Lemma A.5
  $\widetilde{V} \leftarrow S_V^\top V$ ▷ $\widetilde{V} \in \mathbb{R}^{s_V \times d}$
  $S_E \leftarrow$ QNyströmKernel($Q \cup K, (x_i, x_j) \mapsto \exp(\langle x_i, x_j \rangle), 1/\mathrm{poly}(n), \lambda$)
  ▷ Let $x_1, \ldots, x_{2n}$ denote the dataset $Q \cup K$
  $\widetilde{M} \leftarrow \{S_E(i) S_E(j) \cdot \exp(\langle x_i, x_j \rangle)\}_{(i,j) \in S_E \times S_E}$ ▷ $\widetilde{M} \in \mathbb{R}^{s_E \times s_E}$
  $\widetilde{R} \leftarrow \{S_E(i) S_V(j) \cdot \exp(\langle x_i, x_j \rangle)\}_{(i,j) \in S_E \times S_V}$ ▷ $\widetilde{R} \in \mathbb{R}^{s_E \times s_V}, \widetilde{R} = S_E^\top E \widetilde{S}_V$
  $\widetilde{N} \leftarrow \widetilde{M}^\dagger \widetilde{R}$ ▷ $\widetilde{N} \in \mathbb{R}^{s_E \times s_V}$
  $\widetilde{L} \leftarrow \widetilde{N}\widetilde{V}$ ▷ $\widetilde{L} \in \mathbb{R}^{s_E \times d}$
end procedure

procedure Query($i \in [n]$)
  $b_i \leftarrow$ QRN.Query(i) ▷ Algorithm 2
  $u_i \leftarrow \{S_E(j) \cdot \exp(\langle x_i, x_j \rangle\}_{j \in S_E}$ ▷ $u_i \in \mathbb{R}^{s_E}$
  return $\widetilde{L}^\top u_i / b_i$
end procedure
end data structure

λ	s_λ	$\frac{1}{\epsilon\|E\| + \lambda\sqrt{n}}$	β
$\uparrow$	$\downarrow$	$\downarrow$	$\uparrow$
$\downarrow$	$\uparrow$	$\uparrow$	$\downarrow$

Table 1: Parameters s_λ, $\frac{1}{\epsilon\|E\| + \lambda\sqrt{n}}$, and β as functions of λ.

that our approximation of A is obtained via Nyström approximation on the exponential kernel matrix over the rows of Q and K. While this approach yields direct bounds on approximating $\exp(QQ^\top)$ and $\exp(KK^\top)$, it only provides guarantees on $A + A^\top$. To obtain a bound without symmetrization, one would need an alternative method that does not rely on the kernel matrix E.

4 Related Work

Transformers and Attention Mechanism. Transformers (Vaswani et al., 2017) have been the driving force behind large language models (Devlin et al., 2019; Brown et al., 2020; Touvron et al., 2023; Bubeck et al., 2023; Team et al., 2023; Liu et al., 2024a). They are sequence-to-sequence generative models, where the sequence length is typically denoted by n. The key architectural component that distinguishes transformers from earlier models is the attention mechanism, which computes a softmax over the pairwise interactions of query-key vectors. However, computing the full softmax distribution requires $\Omega(n^2)$ time, due to the size of the attention matrix. This quadratic dependency renders transformers inefficient for long sequences, motivating a rich body of work aimed at approximating attention in subquadratic time. These approaches can be broadly categorized into three main classes: (1) *Pattern-based sparse attention*: only a subset of attention matrix entries are computed, with the subset determined by predefined patterns, such as sliding windows or graph-based sparsity structures (Daras et al., 2020; Kitaev et al., 2020; Roy et al., 2021; Sun et al., 2022; Child et al., 2019; Beltagy et al., 2020; Ainslie et al., 2020; Zaheer et al., 2020). (2) *Kernel-based linear attention*: these methods attempt to linearize the kernel by exploiting the identity $\mathsf{K}(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ for a feature map $\phi : \mathbb{R}^d \to \mathbb{R}^m$. When the kernel is exponential, exact computation requires $m = \infty$, so many heuristic approximations for ϕ have been proposed (Katharopoulos et al., 2020; Choromanski et al., 2021; Wang et al., 2020; Peng et al., 2021) with $m = O(d)$. (3) *Data structure-based attention*: these works design specialized data structures for approximating various components of attention. Examples include estimating the normalization factor via kernel density estimation (KDE) (Zandieh et al., 2023), using hashing to identify large entries (Han et al., 2024), applying polynomial approximation methods under bounded input conditions (Alman & Song, 2023), and other algorithmic innovations (Kacham et al., 2024; Zandieh et al., 2024; van den Brand et al., 2024; Song et al., 2024; Kannan et al., 2025; Chu et al., 2024; Chen et al., 2025b; Indyk et al., 2025). Our work falls into the third category, as we design quantum data structures to approximate each of the matrices involved in the attention computation.

Quantum Machine Learning. Given a machine learning problem, can we solve it faster on a quantum computer? The paradigm of using quantum mechanics to accelerate machine learning algorithms has sparked significant interest, leading to a wide array of results across diverse problem domains, including clustering (Kerenidis et al., 2019; Xue et al., 2023), classification (Li et al., 2019), regression (Chen & de Wolf, 2023), training neural networks (Chakrabarti et al., 2019; Kerenidis et al., 2020), convex optimization (Chakrabarti et al., 2020; van Apeldoorn et al., 2020a; Li & Zhang, 2022; Sidford & Zhang, 2023; Zhang et al., 2024b; Wang et al., 2024), mathematical programming (Brandão et al., 2019; van Apeldoorn et al., 2020b; van Apeldoorn & Gilyén, 2019; Kerenidis & Prakash, 2020; Kerenidis et al., 2021; van Apeldoorn et al., 2021; Apers & Gribling, 2023), graph sparsification (Apers & De Wolf, 2022), and recommender systems (Kerenidis & Prakash, 2017). Among the key quantum techniques, Grover search (Grover, 1996) plays a foundational role. It provides a quadratic speedup for database search problems: given a function $f : [n] \to \{0, 1\}$, the goal is to list up to m indices i such that $f(i) = 1$. The Grover search algorithm requires oracle access to f and can produce these m indices using only $O(\sqrt{mn})$ queries, in contrast to the $O(n)$ queries required classically. Several variants of Grover search have been developed to suit different computational settings. In this paper, we use the probabilistic version: given a list of n probabilities $p_1, \ldots, p_n \in [0, 1]$, Grover search can be used to sample a list of indices where each i is selected independently with probability p_i. By the standard analysis of Grover search, this sampling requires $\widetilde{O}(\sqrt{nP})$ queries to the probability values p_i where $P = \sum_{i=1}^n p_i$. Before our work, Gao et al. (2023) also applied Grover search to accelerate attention computation. However, their method requires a structural assumption: for each query $q_i \in \mathbb{R}^d$, the associated set $S_i = \{j \in [n] : \langle q_i, k_j \rangle \geq \tau\}$ must have cardinality at most k. Under this assumption, their algorithm runs in time $\widetilde{O}(n^{1.5} k^{0.5} d + nkd)$. Notably, if $k = n$, then their algorithm offers no speedup over the exact computation.

5 Conclusion

We consider the problem of approximating the attention module in the row query model, where the goal is to return individual rows of the approximate attention matrix. We design a quantum data structure that preprocesses Q, K, and V in $\widetilde{O}(\epsilon^{-1} n^{0.5} \operatorname{poly}(s_\lambda, d, \alpha))$ time, and answers any row query in $\widetilde{O}(s_\lambda^2 + s_\lambda d)$ time. To the best of our knowledge, this is the first quantum algorithm to achieve sublinear dependence on n even in the row query model.

Our work also has several limitations, which raise interesting open questions. First, the error guarantee we obtain is in Frobenius norm rather than spectral norm. While Frobenius norm bounds the sum of the squared ℓ_2 errors across all rows, the spectral norm provides a worst-case guarantee that each row is well approximated. Therefore, it would be desirable to strengthen the result to achieve a spectral norm guarantee. Second, our current guarantee is expressed in terms of the *symmetrization* of the attention matrix. While somewhat unnatural, this is a consequence of approximating the attention matrix through the Nyström method applied to the exponential kernel matrix over the combined dataset $Q \cup K$, where the attention matrix appears as the off-diagonal block. A natural open problem is whether one can obtain approximation guarantees for the attention matrix directly without symmetrization, while still benefiting from quantum speedup in the construction.

Ethics Statement

Our work is a theoretical quantum framework to approximate the attention module in sublinear time. We don't foresee any potential ethics concerns.

Reproducibility Statement

We include all the proofs in the appendix. For proofs of the exponential kernel, see Section A, for proofs of estimating the normalization factor, see Section B. For proofs of the leverage score approximate matrix multiplication, see Section C. The final conclusion is proved in Section D.

References

Muhammad Adnan, Akhil Arunkumar, Gaurav Jain, Prashant J. Nair, Ilya Soloveychik, and Purushotham Kamath. Keyformer: Kv cache reduction through key tokens selection for efficient generative inference. In *Proceedings of the 7th Conference on Machine Learning and Systems (MLSys)*, 2024.

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. ETC: Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 268–284. Association for Computational Linguistics, 2020.

Ahmed El Alaoui and Michael W. Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems 28 (NeurIPS 2015)*, pp. 775–783, 2015.

Josh Alman and Zhao Song. Fast attention requires bounded entries. *Advances in Neural Information Processing Systems*, 36:63117–63135, 2023.

Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More asymmetry yields faster matrix multiplication. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2005–2039, 2025.

Simon Apers and Ronald De Wolf. Quantum speedup for graph sparsification, cut approximation, and laplacian solving. *SIAM Journal on Computing*, 51(6):1703–1742, 2022.

Simon Apers and Sander Gribling. Quantum speedups for linear programming via interior point methods. *arXiv preprint arXiv:2311.03215*, 2023.

Payman Behnam, Yaosheng Fu, Ritchie Zhao, Po-An Tsai, Zhiding Yu, and Alexey Tumanov. Rocketkv: Accelerating long-context llm inference via two-stage kv cache compression. *arXiv preprint arXiv:2502.14051*, 2025.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.

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 (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 27:1–27:14. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2019.

William Brandon, Mayank Mishra, Aniruddha Nrusimha, Rameswar Panda, and Jonathan Ragan-Kelley. Reducing transformer key-value cache size with cross-layer attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott

Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023. URL https://arxiv.org/abs/2303.12712.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 213–229. Springer, 2020.

Shouvanik Chakrabarti, Yiming Huang, Tongyang Li, Soheil Feizi, and Xiaodi Wu. Quantum wasserstein generative adversarial networks. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, pp. 6768–6779, 2019.

Shouvanik Chakrabarti, Andrew M. Childs, Tongyang Li, and Xiaodi Wu. Quantum algorithms and lower bounds for convex optimization. *Quantum*, 4:221, 2020.

Bo Chen, Xiaoyu Li, Yekun Ke, Yingyu Liang, Zhenmei Shi, and Zhao Song. Exploring the limits of kv cache compression in visual autoregressive transformers. *arXiv preprint arXiv:2503.14881*, 2025a.

Bo Chen, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, and Zhao Song. HSR-enhanced sparse attention acceleration. In *The Second Conference on Parsimony and Learning (Proceedings Track)*, 2025b. URL https://openreview.net/forum?id=wso1gABiPZ.

Yanlin Chen and Ronald de Wolf. Quantum algorithms and lower bounds for linear regression with norm constraints. In *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 38:1–38:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

Yifang Chen, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, Zhao Song, and Yu Tian. Time and memory trade-off of kv-cache compression in tensor transformer decoding. *arXiv preprint arXiv:2503.11108*, 2025c.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. *Advances in neural information processing systems*, 28, 2015.

Timothy Chu, Josh Alman, Gary L Miller, Shyam Narayanan, Mark Sellke, and Zhao Song. Metric transforms and low rank representations of kernels for fast attention. *Advances in Neural Information Processing Systems*, 37:47014–47068, 2024.

Kenneth L Clarkson and David P Woodruff. Low-rank approximation and regression in input sparsity time. *Journal of the ACM (JACM)*, 63(6):1–45, 2017.

Arjan Cornelissen, Yassine Hamoudi, and Sofiene Jerbi. Near-optimal quantum algorithms for multivariate mean estimation. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pp. 33–43, New York, NY, USA, 2022. Association for Computing Machinery.

Giannis Daras, Nikita Kitaev, Augustus Odena, and Alexandros G. Dimakis. Smyrf: Efficient attention using asymmetric clustering. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6470–6481, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Ran Duan, Hongxun Wu, and Renfei Zhou. Faster matrix multiplication via asymmetric hashing. In *FOCS*, 2023.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S. Kevin Zhou. Identify critical kv cache in llm inference from an output perturbation perspective. *arXiv preprint arXiv:2502.03805*, 2025.

Yeqi Gao, Zhao Song, Xin Yang, and Ruizhe Zhang. Fast quantum algorithm for attention computation, 2023.

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

Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R Martin, Ming-Ming Cheng, and Shi-Min Hu. Attention mechanisms in computer vision: A survey. *Computational Visual Media*, 8(3):331–368, 2022.

Insu Han, Rajesh Jayaram, Amin Karbasi, Vahab Mirrokni, David P. Woodruff, and Amir Zandieh. Hyperattention: Long-context attention in near-linear time. In *International Conference on Learning Representations (ICLR)*, 2024.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2nd edition, 2009.

Piotr Indyk, Michael Kapralov, Kshiteej Sheth, and Tal Wagner. Improved algorithms for kernel matrix-vector multiplication under sparsity assumptions. In *Proceedings of the International Conference on Learning Representations (ICLR)*, May 2025.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4163–4174. Association for Computational Linguistics, 2020.

Praneeth Kacham, Vahab Mirrokni, and Peilin Zhong. Polysketchformer: Fast transformers via sketching polynomial kernels. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Ravindran Kannan, Chiranjib Bhattacharyya, Praneeth Kacham, and David P. Woodruff. Levattention: Time, space, and streaming efficient algorithm for heavy attentions. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 5156–5165. PMLR, 2020.

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

Iordanis Kerenidis and Anupam Prakash. A quantum interior point method for lps and sdps. *ACM Transactions on Quantum Computing*, 1(1):1–32, 2020.

Iordanis Kerenidis, Jonas Landman, Alessandro Luongo, and Anupam Prakash. q-means: A quantum algorithm for unsupervised machine learning. In *Advances in Neural Information Processing Systems*, volume 32, pp. 4134–4144, 2019.

Iordanis Kerenidis, Jonas Landman, and Anupam Prakash. Quantum algorithms for deep convolutional neural networks. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

Iordanis Kerenidis, Anupam Prakash, and Dániel Szilágyi. Quantum algorithms for Second-Order Cone Programming and Support Vector Machines. *Quantum*, 5:427, 2021.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020.

Lilly Kumari, Shengjie Wang, Tianyi Zhou, Nikhil Sarda, Anthony Rowe, and Jeff Bilmes. Bumblebee: Dynamic kv-cache streaming submodular summarization for infinite-context transformers. In *Proceedings of the Conference on Learning for Molecules (COLM)*, 2024.

Tongyang Li and Ruizhe Zhang. Quantum speedups of optimizing approximately convex functions with applications to logarithmic regret stochastic convex bandits. In *Advances in Neural Information Processing Systems*, volume 35, pp. 19565–19577, 2022.

Tongyang Li, Shouvanik Chakrabarti, and Xiaodi Wu. Sublinear quantum algorithms for training linear and kernel-based classifiers. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3815–3824. PMLR, June 2019.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Minghui Liu, Tahseen Rabbani, Tony O'Halloran, Ananth Sankaralingam, Mary-Anne Hartley, Brian Gravelle, Furong Huang, Cornelia Fermüller, and Yiannis Aloimonos. Hashevict: A pre-attention kv cache eviction strategy using locality-sensitive hashing. *arXiv preprint arXiv:2412.16187*, 2024b.

Xiaoyun Liu, Daniel Esser, Brandon Wagstaff, Anna Zavodni, Naomi Matsuura, Jonathan Kelly, and Eric Diller. Capsule robot pose and mechanism state detection in ultrasound using attention-based hierarchical deep learning. *Scientific Reports*, 12(1):21130, 2022.

Cameron Musco and Christopher Musco. Recursive sampling for the nyström method. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 3836–3848, Red Hook, NY, USA, 2017. Curran Associates Inc.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A. Smith, and Lingpeng Kong. Random feature attention. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. In *Proceedings of the 6th Conference on Machine Learning and Systems (MLSys)*, 2023.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021.

Aaron Sidford and Chenyi Zhang. Quantum speedups for stochastic optimization. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, pp. 1–12, 2023.

Zhao Song, Junze Yin, and Lichen Zhang. Solving attention kernel regression problem via pre-conditioner. In *International Conference on Artificial Intelligence and Statistics*, pp. 208–216. PMLR, 2024.

Zhiqing Sun, Yiming Yang, and Shinjae Yoo. Sparse attention with learning to hash. In *International Conference on Learning Representations*, 2022.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Joran van Apeldoorn and András Gilyén. Improvements in quantum sdp-solving with applications. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 99:1–99:15. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2019.

Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Convex optimization using quantum oracles. *Quantum*, 4:220, 2020a.

Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Quantum sdp-solvers: Better upper and lower bounds. *Quantum*, 4:230, 2020b.

Joran van Apeldoorn, Sander Gribling, Yinan Li, Harold Nieuwboer, Michael Walter, and Ronald de Wolf. Quantum algorithms for matrix scaling and matrix balancing. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 110:1–110:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.

Jan van den Brand, Zhao Song, and Tianyi Zhou. Algorithm and hardness for dynamic attention maintenance in large language models. In *International Conference on Machine Learning*, pp. 49008–49028. PMLR, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Hao Wang, Chenyi Zhang, and Tongyang Li. Near-optimal quantum algorithm for minimizing the maximal loss. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity, 2020.

Yuanchao Wang, Wenji Du, Chenghao Cai, and Yanyan Xu. Explaining the attention mechanism of end-to-end speech recognition using decision trees. *arXiv preprint arXiv:2110.03879*, 2021.

Christopher K. I. Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Proceedings of the 14th International Conference on Neural Information Processing Systems*, NIPS'00, pp. 661–667, Cambridge, MA, USA, 2000. MIT Press.

Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 3792–3835. SIAM, 2024.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *AAAI*, pp. 14138–14148. AAAI Press, 2021.

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32, pp. 5754–5764. Curran Associates, Inc., 2019.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17283–17297, 2020.

Amir Zandieh, Insu Han, Majid Daliri, and Amin Karbasi. Kdeformer: Accelerating transformers via kernel density estimation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40605–40623. PMLR, 2023.

Amir Zandieh, Insu Han, Vahab Mirrokni, and Amin Karbasi. Subgen: Token generation in sublinear time and memory. *arXiv preprint arXiv:2402.06082*, 2024.

Rongzhi Zhang, Kuang Wang, Liyuan Liu, Shuohang Wang, Hao Cheng, Chao Zhang, and Yelong Shen. Lorc: Low-rank compression for llms kv cache with a progressive compression strategy. In *NeurIPS 2024 Workshop on Model Compression*, 2024a.

Tong Zhang. Learning bounds for kernel regression using effective data dimensionality. *Neural Computation*, 17(9):2077–2098, 2005.

Yexin Zhang, Chenyi Zhang, Cong Fang, Liwei Wang, and Tongyang Li. Quantum algorithms and lower bounds for finite-sum optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12345–12356. PMLR, 2024b.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115. AAAI Press, 2021.

Appendix

Roadmap. In Section A, we describe the quantum algorithm for exponential kernels. In Section B, we discuss how to estimate the normalization factor. In Section C, we show the details on approximating matrix multiplication via leverage scores. In Section D, we combine things together and obtain the main result.

A Quantum Algorithm for Exponential Kernel

In this section, we give a generic reduction from attention matrix to a kernel matrix. Given queries and keys $Q = \{q_1, \dots, q_n\}, K = \{k_1, \dots, k_n\}$, recall that we are interested in the matrix $\exp(QK^\top)$ where the (i, j)-th entry is $\exp(q_i^\top k_j)$, and this matrix is not a PSD kernel matrix. We show a reduction that first computes the exponential kernel $\mathsf{K}(x, y) = \exp(\langle x, y\rangle)$ over the dataset $Q \cup K$, then we can effectively extract certain blocks of the kernel matrix E that approximates $\exp(QK^\top)$ well. We start with a lemma on block approximation.

Lemma A.1. *Let $E \in \mathbb{R}^{2n\times 2n}$ be a PSD matrix and $E = \begin{bmatrix} B & A \\ A^\top & C \end{bmatrix}$ where each block is of size $n \times n$. Suppose there exists a matrix $\widetilde{E} \in \mathbb{R}^{2n\times 2n}$ such that $E \preceq \widetilde{E} \preceq E + \lambda I$ for $\lambda > 0$ and let $\widetilde{E} = \begin{bmatrix} \widetilde{B} & \widetilde{A} \\ \widetilde{A}^\top & \widetilde{C} \end{bmatrix}$, then we have*

$$A + A^\top - 2\lambda I \preceq \widetilde{A} + \widetilde{A}^\top \preceq A + A^\top.$$

Proof. We first note that since $\widetilde{E}$ spectrally approximates E, so do $\widetilde{B}$ approximate B and $\widetilde{C}$ approximate C. Let $x \in \mathbb{R}^n$ and consider $x_1 = \begin{bmatrix} x \\ \mathbf{0}_n \end{bmatrix}$ and $x_2 = \begin{bmatrix} \mathbf{0}_n \\ x \end{bmatrix}$, then

$$x_1^\top E x_1 = x^\top B x,$$
$$x_2^\top E x_2 = x^\top C x,$$

therefore we have

$$B \preceq \widetilde{B} \preceq B + \lambda I,$$
$$C \preceq \widetilde{C} \preceq C + \lambda I.$$

Now, set $x_3 = x_1 + x_2$, and we compute the quadratic form:

$$\begin{aligned} x_3^\top E x_3 &= (x_1 + x_2)^\top E (x_1 + x_2) \\ &= x_1^\top E x_1 + x_2^\top E x_2 + x_1^\top E x_2 + x_2^\top K x_1 \\ &= x^\top B x + x^\top C x + x^\top (A + A^\top) x, \end{aligned}$$

by the spectral approximation guarantee, we have

$$B + C + (A + A^\top) \preceq \widetilde{B} + \widetilde{C} + (\widetilde{A} + \widetilde{A}^\top) \preceq B + C + (A + A^\top) + \lambda I, \tag{1}$$

finally use the guarantees on $\widetilde{B}, \widetilde{C}$ yields,

$$-\lambda I \preceq B - \widetilde{B} \preceq 0,$$
$$-\lambda I \preceq C - \widetilde{C} \preceq 0,$$

incorporate these bounds into Eq. (1), we conclude

$$A + A^\top - 2\lambda I \preceq \widetilde{A} + \widetilde{A}^\top \preceq A + A^\top,$$

as desired. This completes the proof. □

Our plan is to form the kernel matrix over the dataset $Q \cup K$ implicitly via Nyström approximation, then extract corresponding blocks to approximate $\exp(QK^\top)$.

Corollary A.2. *Let $Q, K \in \mathbb{R}^{n\times d}$ and let $E \in \mathbb{R}^{2n\times 2n}$ be the exponential kernel matrix over the dataset $Q \cup K$, suppose there exists an $\widetilde{E} \in \mathbb{R}^{2n\times 2n}$ such that $E \preceq \widetilde{E} \preceq E + \lambda I$ for some $\lambda > 0$, then there exists $\widetilde{B} \in \mathbb{R}^{n\times n}$ such that*

$$\exp(QK^\top) + \exp(KQ^\top) - 2\lambda I \preceq \widetilde{A} + \widetilde{A}^\top \preceq \exp(QK^\top) + \exp(KQ^\top)$$

Proof. The result is a consequence of Lemma A.1 by identifying that

$$E = \begin{bmatrix} \exp(QQ^\top) & \exp(QK^\top) \\ \exp(KQ^\top) & \exp(KK^\top) \end{bmatrix},$$

and $\widetilde{E}$ contains proper approximations for the desired blocks. □

It remains to give an efficient algorithm to approximate the exponential kernel matrix E. A popular scheme is via Nyström approximation (Williams & Seeger, 2000): the algorithm selects a subset of "landmark" points, and constructs $\widetilde{E}$ through these landmarks. Musco & Musco (2017) uses recursive ridge leverage score sampling to generate such an approximation efficiently. Musco & Musco (2017) presents an algorithm that uses $\widetilde{O}(ns_\lambda \log(1/\delta))$ kernel function evaluations and $\widetilde{O}(ns_\lambda^2 \log(1/\delta))$ additional runtime to compute an approximation $\widetilde{K}$ satisfying $K \preceq \widetilde{K} \preceq K + \lambda I$ with probability at least $1-\delta$. We restate their main result here for the sake of completeness.

Lemma A.3 (Theorem 7 of Musco & Musco (2017))**.** *Let $s = O(s_\lambda \log(s_\lambda/\delta))$, there exists a weighted sampling matrix $S \in \mathbb{R}^{n\times s}$, such that the Nyström approximation of E, $\widetilde{E} = ES(S^\top ES)^\dagger S^\top E$ satisfies*

$$E \preceq \widetilde{E} \preceq E + \lambda I,$$

holds with probability at least $1-\delta$. Moreover, S can be computed using $O(ns)$ kernel evaluations and $O(ns^2)$ additional time.

Our main contribution is a quantum algorithm that generates the approximation in *sublinear time*. Before introducing the algorithm, we recall several key concepts.

Lemma A.3 relies on approximating the ridge leverage score on a sample, which can be captured by the notion of generalized ridge leverage score.

Definition A.4 (Generalized ridge leverage score, Musco & Musco (2017))**.** *Let $E \in \mathbb{R}^{n\times n}$ be a kernel matrix, let $\lambda > 0$, and let $S \in \mathbb{R}^{n\times s}$ be any weighted sampling matrix, the λ-generalized ridge leverage score with respect to S, is defined for any $i \in [n]$,*

$$\widetilde{\tau}_i^\lambda := \frac{1}{\lambda}(E - ES(S^\top ES + \lambda I)^{-1} S^\top E)_{i,i},$$

let $B \in \mathbb{R}^{n\times n}$ be any factorization of $E = BB^\top$, it can be equivalently defined as

$$\widetilde{\tau}_i^\lambda = b_i^\top (B^\top S^\top SB + \lambda I)^{-1} b_i,$$

where b_i is the i-th row of B.

We also need a procedure introduced in Apers & Gribling (2023) that generates a spectral approximation of an $n \times d$ matrix, given only queries to its rows, using quantum leverage score sampling. We record it here.

Lemma A.5 (Theorem 3.1 of Apers & Gribling (2023))**.** *Let $U \in \mathbb{R}^{n\times d}$, $\epsilon, \delta \in (0,1)$. There exists a quantum algorithm that computes a weighted sampling matrix $S \in \mathbb{R}^{n\times s}$ with $s = O(\epsilon^{-2} d \log(d/\delta))$ such that with probability at least $1-\delta$,*

$$(1-\epsilon)U^\top U \preceq U^\top SS^\top U \preceq (1+\epsilon)U^\top U.$$

The quantum algorithm uses $\widetilde{O}(\epsilon^{-1} n^{0.5} d^{0.5})$ row queries to U, and it takes time $\widetilde{O}(\epsilon^{-1} n^{0.5} d^{1.5} + d^\omega)$. Moreover, if the leverage score sampling matrix contains $s \le d$ rows, then the algorithm uses $\widetilde{O}(n^{0.5} s^{0.5})$ row queries to U and it takes time $\widetilde{O}(n^{0.5} s^{0.5} d + d^\omega)$. We use $\textsc{QLeverageScore}(U, s)$ to denote this procedure that produces a leverage score sampling matrix $S \in \mathbb{R}^{n\times s}$.

We prove the key algorithmic result of this section.

Theorem A.6. *Let $\{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$ be a dataset, $\mathsf{K} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^m$ be a kernel function, $\lambda > 0$ and $\delta \in (0, 1)$. Let E be the kernel matrix where $E_{i,j} = \mathsf{K}(x_i, x_j)$. Suppose $s = O(s_\lambda \log(s_\lambda/\delta))$, then Algorithm 1 computes a weighted sampling matrix $S \in \mathbb{R}^{n\times s}$ such that with probability at least $1 - \delta$,*

$$E \preceq \widetilde{E} \preceq E + \lambda I,$$

where $\widetilde{E} = ES(S^\top ES)^\dagger S^\top E$. Moreover, S can be computed in time $\widetilde{O}(n^{0.5}s^{1.5} \cdot (\mathcal{T}_\mathsf{K} + s) + s^\omega)$, where $\mathcal{T}_\mathsf{K}$ is the time to evaluate the kernel function.

Proof. We note that the major differences between Algorithm 1 and the algorithm in Musco & Musco (2017) are

- Musco & Musco (2017) algorithm is recursive, our algorithm unrolls the recursion and iteratively constructs the weighted sampling matrix;
- Musco & Musco (2017) computes all p_i's classically, while we use QSAMPLE to generate samples.

Hence, the correctness is automatically satisfied. It remains to give a bound on the running time.

- Computing M_0: $M_0 \in \mathbb{R}^{s\times s}$ contains the values of kernel functions over s^2 pairs, forming it takes $O(s^2) \cdot \mathcal{T}_\mathsf{K}$ time;
- Computing $\widehat{M}$: we maintain the invariant that $M_t \in \mathbb{R}^{s\times s}$ for all $t \in [T]$, therefore computing $\widehat{M}$ is inverting an $s \times s$ matrix, which takes $O(s^\omega)$ time;
- Computing $D_{t-1}^\top K_i$: this operation involves computing s weighted kernel function evaluations, given D_{t-1} stores a list of s indices together with weights, it can be done in $O(s) \cdot \mathcal{T}_\mathsf{K}$ time;
- Oracle for q_i: for any fixed i, note that we need to form $D_{t-1}^\top K_i$ using $O(s) \cdot \mathcal{T}_\mathsf{K}$ time, and computing the quadratic form takes $O(s^2)$ time. Thus each oracle call takes $O(s)\cdot\mathcal{T}_\mathsf{K}+O(s^2)$ time;
- Computing $\widetilde{D}_t$: this step requires to compute at most n probabilities, and each probability can be computed via an oracle call in $O(s) \cdot \mathcal{T}_\mathsf{K} + O(s^2)$ time, so it remains to give a bound on the sum of probabilities. By the definition of p_i,
$$\sum_{i=1}^n p_i \leq 16 \log(2s/\delta) \sum_{i=1}^n q_i,$$
and the sum of q_i's is
$$\begin{aligned}\sum_{i=1}^n q_i &= \frac{5}{\lambda} \cdot (\mathsf{K}(x_i, x_i) - (D_{t-1}^\top K_i)^\top \widehat{M} (D_{t-1}^\top K_i)) \\ &= \frac{5}{\lambda} \cdot (E - ED_{t-1}(D_{t-1}^\top ED_{t-1} + \lambda I)^{-1} D_{t-1}^\top E)_{i,i} \\ &= 5 \cdot \sum_{i=1}^n \widetilde{\tau}_i^\lambda,\end{aligned}$$
by Theorem 8 of Musco & Musco (2017), the sum of λ-generalized ridge leverage score with sampling matrix D_{t-1} is at most $O(s_\lambda \log(s_\lambda/\delta)) = s$, thus the runtime is $\widetilde{O}(n^{0.5}s^{1.5} \cdot (\mathcal{T}_\mathsf{K} + s))$.

Finally, note that the loop is dominated by the last iteration, and at each iteration, the number of points to consider is divided by half, we conclude the overall runtime of Algorithm 1 is

$$\widetilde{O}(n^{0.5}s^{1.5} \cdot (\mathcal{T}_\mathsf{K} + s) + s^\omega),$$

as desired. $\square$

We can then apply Theorem A.6 to exponential kernel function and the dataset $Q \cup K$ to compute a Nyström sampling matrix S.

Corollary A.7. *Let $Q, K \in \mathbb{R}^{n\times d}$, $\lambda > 0$ and $\delta \in (0,1)$. Define the dataset $X = \{x_1, x_2, \ldots, x_{2n}\} \subseteq \mathbb{R}^d$ where for $i \in [n]$, $x_i = q_i$ and for $i \in \{n+1, \ldots, 2n\}$, $x_i = k_i$. Let E be the kernel matrix where $E_{i,j} = \exp(\langle x_i, x_j\rangle)$. Suppose $s = O(s_\lambda \log(s_\lambda/\delta))$, then there exists an algorithm that computes a weighted sampling matrix $S \in \mathbb{R}^{2n\times s}$ such that, let $\widetilde{E} = ES(S^\top ES)^\dagger S^\top E$, then with probability at least $1-\delta$, $E \preceq \widetilde{E} \preceq E + \lambda I$. Moreover, S can be computed in $\widetilde{O}(n^{0.5}s^{1.5}\cdot(d+s) + s^\omega)$ time.*

Proof. Apply Theorem A.6 to the kernel function $\mathsf{K}(x_i, x_j) = \exp(\langle x_i, x_j\rangle)$ and note that the kernel function can be computed in $O(d)$ time. □

B Estimating the Normalization Factor

Given a sublinear quantum algorithm to approximate the matrix $\exp(QK^\top)$, our next step is to estimate the normalization factor $\exp(QK^\top)\mathbf{1}_n$ to compute the softmax matrix. We first show that given a Nyström approximation to the $2n \times 2n$ kernel matrix E, how to compute the normalization factor and the approximate guarantees.

Lemma B.1. *Let $M \in \mathbb{R}^{n\times n}$ be a symmetric matrix, then we have*

$$\|M\|_\infty \leq \sqrt{n}\cdot\|M\|.$$

Proof. Fix any $i \in [n]$, we examine the row $M_{i,*}$, set the test vector x to be $x_j = \begin{cases} +1, & \text{if } M_{i,j} \geq 0, \\ -1, & \text{otherwise.}\end{cases}$, then

$$\begin{aligned}
\|M_{i,*}\|_1 &= M_{i,*}^\top x \\
&= \langle Me_i, x\rangle \\
&\leq \|Me_i\|_2 \cdot \|x\|_2 \\
&\leq \|M\| \cdot \|x\|_2 \\
&= \sqrt{n}\cdot\|M\|.
\end{aligned}$$

The conclusion can be achieved by noting that this bound works for any row i. □

There are two major issues for estimating the normalization factor:

- Corollary A.7 only allows us to compute the sampling matrix in sublinear time, explicitly forming the Nyström approximation $\widetilde{E}$ however, would require $\Omega(n)$ time since the matrix is of size $n \times n$;
- Even though we are given the explicit factorization $\widetilde{E} = UU^\top$ where $U \in \mathbb{R}^{2n\times s}$, we would have to compute n normalization factors, which would require $\Omega(n)$ time.

In other words, because the output has size $\Omega(n)$, one cannot expect any quantum algorithm that runs in $o(n)$ time. Instead, we design a quantum data structure with preprocessing time $o(n)$ time, and can support query to compute the normalization factor to any row efficiently.

In particular, we are interested in the following algorithmic task: given query access to the rows of a matrix $U \in \mathbb{R}^{n\times s}$ and a vector $v \in \mathbb{R}^n$, output a vector $\widetilde{\mu} \in \mathbb{R}^s$ such that $\|\widetilde{\mu} - U^\top v\|_{(U^\top U)^{-1}} \leq \epsilon$, which can be solved via Lemma 2.6. For our application, $\|v\|_\infty = 1$. However, we are interested in the quantity $UU^\top v$ so we need to measure the error $\|U(\widetilde{\mu} - U^\top v)\|_2$. How would a bound on the $\|\cdot\|_{(U^\top U)^{-1}}$ be useful? We prove a structural lemma below.

Lemma B.2. *Let $M \in \mathbb{R}^{s\times s}$ be a PD matrix and $x \in \mathbb{R}^s$ satisfy $\|x\|_{M^{-1}} \leq \epsilon$ for some $\epsilon \in (0,1)$. Then, we have*

$$\|x\|_M \leq \epsilon\cdot\|M\|.$$

Proof. Let $M = U\Lambda U^\top$ be its eigendecomposition, we consider the squared norm of x:

$$\begin{aligned}
\|x\|_{M^{-1}}^2 &= x^\top M^{-1} x \\
&= x^\top U \Lambda^{-1} \underbrace{U^\top x}_{y} \\
&= y^\top \Lambda^{-1} y \\
&= \sum_{i=1}^{s} \frac{1}{\lambda_i} y_i^2,
\end{aligned}$$

meanwhile, by the same token of argument, we have

$$\begin{aligned}
\|x\|_M^2 &= y^\top \Lambda y \\
&= \sum_{i=1}^{s} \lambda_i y_i^2 \\
&= \sum_{i=1}^{s} \lambda_i^2 \cdot \frac{1}{\lambda_i} y_i^2 \\
&\leq \lambda_{\max}^2 \sum_{i=1}^{s} \frac{1}{\lambda_i} y_i^2 \\
&= \|M\|^2 \cdot \|x\|_{M^{-1}}^2 \\
&\leq \epsilon^2 \cdot \|M\|^2,
\end{aligned}$$

this concludes the proof. □

Corollary B.3. *Let $\delta \in (0, 1)$, $U \in \mathbb{R}^{n\times s}$ and $v \in \mathbb{R}^n$, suppose there exists a vector $\widetilde{\mu} \in \mathbb{R}^s$ with $\|\widetilde{\mu} - U^\top v\|_{(U^\top U)^{-1}} \leq \epsilon$, then we have*

$$\|UU^\top v - U\widetilde{\mu}\|_2 \leq \epsilon \cdot \|U^\top U\|.$$

Proof. Let $M = U^\top U$ and $x = \widetilde{\mu} - U^\top v$, then note that

$$\begin{aligned}
\|U(U^\top v - \widetilde{\mu})\|_2^2 &= \|Ux\|_2^2 \\
&= x^\top U^\top U x \\
&= x^\top M x \\
&= \|x\|_M^2 \\
&\leq \epsilon^2 \cdot \|M\|^2,
\end{aligned}$$

where the last step is by Lemma B.2. □

We are now in the position to state our formal theorem, which provides an end-to-end guarantee on estimating the normalization factor. For simplicity, we will prove the statement with high probability guarantee, i.e., the success probability is $1 - 1/\operatorname{poly}(n)$.

Theorem B.4. *Let $Q, K \in \mathbb{R}^{n\times d}$, $\lambda > 0$ and $\epsilon \in (0, 1)$. Let $s = \widetilde{O}(s_\lambda)$ where s_λ is the statistical dimension of the exponential kernel on $Q \cup K$. There exists a data structure (Algorithm 2) with the following specification:*

- *Preprocessing in time $\widetilde{O}(n^{0.5} s^{1.5} (s + d)/\epsilon + s^\omega)$;*
- *For any $i \in [n]$, it outputs an approximate normalization factor for row i in time $O(s(s+d))$.*

Moreover, with probability at least $1 - 1/\operatorname{poly}(n)$, it holds that for any $i \in [n]$, the output b_i satisfies

$$|b_i - (\exp(q_i K^\top)\mathbf{1}_n| \leq O(\epsilon\|E\| + \lambda\sqrt{n}),$$

if $\frac{\lambda\sqrt{n}}{\|E\|} \leq 1$*, then the bound can be further simplified to*

$$|b_i - \exp(q_i K^\top)\mathbf{1}_n| \leq O(\lambda\sqrt{n}),$$

and the preprocessing time simplifies to

$$\widetilde{O}(s^{1.5}(s+d)\|E\|/\lambda + s^\omega).$$

Proof. Given Q, K, let $E \in \mathbb{R}^{2n\times 2n}$ be the associated exponential kernel matrix. We will first invoke Corollary A.7 to compute a sampling matrix $S \in \mathbb{R}^{2n\times s}$ where $s = \widetilde{O}(s_\lambda)$ such that $\widetilde{E} = ES(S^\top ES)^\dagger S^\top E$ approximates E, in time $\widetilde{O}(n^{0.5}s^{1.5}(s+d)+s^\omega)$. Set $U = ES(S^\top ES)^{\dagger/2}$, we have that $\widetilde{E} = UU^\top$. Note that forming U explicitly would take $\Omega(n)$ time, so we instead implement a row oracle for U. Since $U \in \mathbb{R}^{n\times s}$, we only need to compute s entries for each row, and let $N = (S^\top ES)^{\dagger/2}$, we see that $u_j = N(ES)_{j,*}$ and $(ES)_{j,*}$ contains values in the form of $S_k \cdot \exp(\langle x_j, x_k\rangle)$ for $k \in S$. N can be computed in $O(s^2 d + s^\omega)$ time, and row oracle for any $j \in [n]$ can be implemented in $O(sd + s^2)$ time. By Lemma 2.6, $\widetilde{\mu}$ can be computed in $\widetilde{O}(n^{0.5}s^{1.5}(s+d)/\epsilon)$ time. To query the normalization factor for row i, note that it can be computed via $(U\widetilde{\mu})_i = \langle u_i, \widetilde{\mu}\rangle$, which can be computed using row oracle, in $O(s(s+d))$ time. Thus, the overall runtime of our procedure can be summarized as

- Preprocessing time $\widetilde{O}(n^{0.5}s^{1.5}(s+d)/\epsilon + s^\omega)$;
- Query time $O(s(s+d))$.

It remains to give an approximation guarantee. With probability at least $1 - 1/\operatorname{poly}(n)$, we have $E \preceq \widetilde{E} \preceq E + \lambda I$, and observe that

$$\begin{aligned} |\widetilde{a}_i^\top \mathbf{1_n} - \exp(q_i K^\top)\mathbf{1}_n| &\leq \|(\widetilde{E} - E)v\|_\infty \\ &\leq \|\widetilde{E} - E\|_\infty \cdot \|v\|_\infty \\ &\leq \lambda\sqrt{n}, \end{aligned}$$

where the second step is by the matrix infinity norm is the induced norm of vector ℓ_∞ norm, and the last step is by Lemma B.1. On the other hand, our final output b_i is an approximation to $\widetilde{a}_i^\top \mathbf{1}_n$. Let $\widetilde{y} := U\widetilde{\mu}$, by Corollary B.3, we have

$$\|\widetilde{E}v - \widetilde{y}\|_2 \leq \epsilon \cdot \|U^\top U\|,$$

this holds with probability at least $1-\delta$, conditioning on this event, and note that $\widetilde{E} = UU^\top$, therefore $\|U^\top U\| = \|\widetilde{E}\|$ because $UU^\top$ and $U^\top U$ have the same spectrum. Since $E \preceq \widetilde{E} \preceq E + \lambda I$, we could bound the spectral norm of $\widetilde{E}$ as $\|\widetilde{E}\| \leq \|E\| + \lambda$ simply by triangle inequality. Thus, we conclude our final result by

$$\begin{aligned} |b_i - \exp(q_i K^\top)\mathbf{1}_n| &\leq |b_i - \widetilde{a}_i \mathbf{1}_n| + |\widetilde{a}_i^\top \mathbf{1}_n - \exp(q_i K^\top)\mathbf{1}_n| \\ &\leq \|\widetilde{E}v - \widetilde{y}\|_2 + \lambda\sqrt{n} \\ &\leq \epsilon \cdot (\lambda + \|E\|) + \lambda\sqrt{n}. \end{aligned}$$

Now, suppose $\lambda\sqrt{n} \leq \|E\|$, then we could set $\epsilon = \frac{\lambda\sqrt{n}}{\|E\|}$, the error bound simplifies to $O(\lambda\sqrt{n})$. □

C APPROXIMATE MATRIX MULTIPLICATION VIA LEVERAGE SCORE

It remains to handle the value matrix, and we will do so via a machinery called approximate matrix multiplication.

Definition C.1 (Approximate matrix multiplication, Clarkson & Woodruff (2017))**.** *Let* $A \in \mathbb{R}^{n\times d}, B \in \mathbb{R}^{n\times m}$ *and let* $C = A^\top B \in \mathbb{R}^{d\times m}$*. The approximate matrix multiplication problem asks to design a random matrix* $S \in \mathbb{R}^{n\times s}$*, such that*

$$\Pr[\|A^\top SS^\top B - C\|_F \leq \epsilon \|A\|_F \|B\|_F] \geq 1 - \delta,$$

where $\epsilon, \delta \in (0,1)$*. We call such* S *satisfying* (ϵ, δ)*-AMM.*

To generate the random matrix S, our strategy will be performing leverage score sampling over V. However, standard proof (see, e.g., Clarkson & Woodruff (2017)) requires V to have orthonormal columns. We provide a proof for the case where V does not have orthonormal columns (albeit it requires extra factors in blowups). Before doing so, we define a parameter that quantifies this blowup which we call *row distortion*.

Definition C.2 (Row distortion). *Let $A \in \mathbb{R}^{n\times d}$ for $n \geq d$, we define the row distortion of A, denoted by $\alpha(A)$, as*

$$\alpha(A) := \frac{d}{\|A\|_F^2} \cdot \max_{i\in[n]} \frac{\|a_i\|_2^2}{\tau_i},$$

where a_i is the i-th row of A and τ_i is the i-th leverage score of A (Definition 2.1). When A is clear from context, we use α as an abbreviation.

We are now ready to prove a generalized approximate matrix multiplication based on leverage score sampling, when the matrix does not have orthonormal columns.

Lemma C.3. *Let $A \in \mathbb{R}^{n\times d}$, $B \in \mathbb{R}^{n\times m}$, let $S \in \mathbb{R}^{n\times s}$ be the leverage score sampling matrix of A with $s = (\epsilon^{-2}\alpha \log(1/\delta))$ for $\epsilon, \delta \in (0, 1)$ and α is the row distortion of A (Definition C.2). Then, S is an (ϵ, δ)-AMM.*

Proof. For the sampling matrix S, it is a scaled submatrix of the permutation matrix, where for any $m \in [s]$, $S_{m,z_m} = \frac{1}{\sqrt{sp_m}}$ where $p_m \geq \frac{\tau_m}{d}$ and $z_m = i$ with probability p_i. Let a_i, b_j denote the i-th and j-th row of A and B, respectively. We can write

$$A^\top SS^\top B - A^\top B = \frac{1}{s} \sum_{i\in[n],m\in[s]} a_i b_i^\top \left(\frac{\mathbb{I}[z_m = i]}{p_i} - 1\right),$$

taking expectation, we obtain

$$\begin{aligned}\mathbb{E}[A^\top SS^\top B - A^\top B] &= \frac{1}{s}\sum_{i=1}^n a_i b_i^\top \left(\frac{p_i}{p_i} - 1\right)\\ &= 0,\end{aligned}$$

to bound the second moment of $\|A^\top SS^\top B - A^\top B\|_F$, we first expand the definition of Frobenius norm square:

$$\begin{aligned}&\mathbb{E}\operatorname{tr}[(A^\top SS^\top B - A^\top B)(A^\top SS^\top B - A^\top B)]\\ &= \mathbb{E}\,\frac{1}{s^2}\operatorname{tr}\left[\sum_{i,j\in[n],m\in[s]} b_j a_j^\top a_i b_i^\top \left(\frac{\mathbb{I}[z_m = j]}{p_j} - 1\right)\left(\frac{\mathbb{I}[z_m = i]}{p_i} - 1\right)\right]\\ &= \frac{1}{s^2}\sum_{m=1}^s \operatorname{tr}[\sum_{i=1}^n \frac{1}{p_i}\cdot b_i a_i^\top a_i b_i^\top - B^\top AA^\top B]\\ &= \frac{1}{s}\operatorname{tr}[\sum_{i=1}^n \frac{1}{p_i}\cdot b_i a_i^\top a_i b_i^\top - B^\top AA^\top B]\\ &\leq \frac{1}{s}\left(\sum_{i=1} \frac{1}{p_i}\|a_i\|_2^2\|b_i\|_2^2 - \operatorname{tr}[B^\top AA^\top B]\right)\\ &\leq \frac{1}{s}(\alpha\|A\|_F^2\|B\|_F^2 - \|A^\top B\|_F^2)\\ &\leq \frac{\alpha}{s}\|A\|_F^2\|B\|_F^2,\end{aligned}$$

where the first step is by definition of S, the second step is by applying expectation and use $\mathbb{E}[A^\top SS^\top B - A^\top B] = 0$, the fourth step is by $\operatorname{tr}[b_i a_i^\top a_i b_i^\top] = \|a_i b_i^\top\|_F^2 \leq \|a_i\|_2^2\|b_i\|_2^2$, the fifth step is by $p_i \geq \frac{\tau_i}{d}$, therefore

$$\frac{1}{p_i} \leq \frac{d}{\tau_i}$$

$$= \frac{\|A\|_F^2}{\|a_i\|_2^2} \cdot \frac{d}{\|A\|_F^2} \cdot \frac{\|a_i\|_2^2}{\tau_i}$$
$$\leq \alpha \cdot \frac{\|A\|_F^2}{\|a_i\|_2^2},$$

where the last step is by the definition of α. By Chebyshev's inequality, we can choose $s = O(\alpha/\epsilon^2)$ so that the approximate matrix multiplication holds with constant probability, and one could boost the success probability to $1 - \delta$ by either taking $\log(1/\delta)$ independent copies via a Chernoff bound, or directly through Bernstein inequality. $\square$

We are ready to state our final result on approximating the value matrix V.

Theorem C.4. *Let $V \in \mathbb{R}^{n\times d}$, $\epsilon \in (0, 1)$ and α be the row distortion of V. There exists a quantum algorithm that computes a weighted sampling matrix $S \in \mathbb{R}^{n\times s}$ with $s = \widetilde{O}(\epsilon^{-2}\alpha)$ such that for any fixed matrix $B \in \mathbb{R}^{n\times m}$, S is an $(\epsilon, 1/\operatorname{poly}(n))$-AMM. Moreover, S can be computed using $\widetilde{O}(\epsilon^{-1}n^{0.5}\alpha^{0.5})$ row queries to V and $\widetilde{O}(\epsilon^{-1}n^{0.5}\alpha^{0.5}d + d^\omega)$ time.*

Proof. The proof is by composing Lemma A.5 and Lemma C.3, and note that for $\widetilde{O}(\epsilon^{-2}\alpha)$ rows, the sum of leverage scores is at most $\widetilde{O}(\epsilon^{-2}\alpha)$. $\square$

D PUT THINGS TOGETHER

We are now ready to state our final algorithm and its guarantee. Recall that, we define $D = \exp(QK^\top)\mathbf{1}_n$ and $D' = \exp(KQ^\top)\mathbf{1}_n$. We use $\widetilde{D}, \widetilde{D}'$ to denote their approximations.

We prove a simple inequality that quantifies the perturbation on the inverse.

Lemma D.1. *Let $C, D \in \mathbb{R}^{n\times n}$, if D is nonsingular and $\|C - D\| \leq \epsilon$, and $\|D^{-1}\| < 1/\epsilon$, then C is also nonsingular and $\|C^{-1}\| \leq \frac{\|D^{-1}\|}{1-\epsilon\cdot\|D^{-1}\|}$.*

Proof. We will make use of Neumann series, which states that for $\|A\| < 1$, $(I - A)^{-1}$ admits the expansion

$$(I - A)^{-1} = \sum_{k=0}^{\infty} A^k,$$

this leads to a bound on the norm:

$$\begin{aligned} \|(I - A)^{-1}\| &= \|\sum_{k=0}^{\infty} A^k\| \\ &\leq \sum_{k=0}^{\infty} \|A^k\| \\ &\leq \sum_{k=0}^{\infty} \|A\|^k \\ &= \frac{1}{1 - \|A\|}, \end{aligned} \tag{2}$$

now, to prove our desired bound, we write $C = D + E$ where E is the perturbation, then $C = D + E = D(I + D^{-1}E)$, and we will apply Eq. (2) to $-D^{-1}E$:

$$\begin{aligned} \|D^{-1}E\| &\leq \|D^{-1}\| \cdot \|E\| \\ &= \|D^{-1}\| \cdot \|C - D\| \\ &< 1/\epsilon \cdot \epsilon \\ &= 1, \end{aligned}$$

therefore

$$\begin{aligned}\|C^{-1}\| &= \|(I + D^{-1}E)^{-1}D^{-1}\| \\ &\leq \|D\| \cdot \|(I - D^{-1}E)^{-1}\| \\ &\leq \frac{\|D^{-1}\|}{1 - \|D^{-1}E\|} \\ &\leq \frac{\|D^{-1}\|}{1 - \|E\| \cdot \|D^{-1}\|} \\ &\leq \frac{\|D^{-1}\|}{1 - \epsilon \cdot \|D^{-1}\|},\end{aligned}$$

this completes the proof. $\square$

Theorem D.2 (Formal version of Theorem 3.1)**.** *Let $Q, K, V \in \mathbb{R}^{n\times d}$ be the query, key and value matrices for attention, let $\epsilon, \lambda > 0$. Let $E \in \mathbb{R}^{2n\times 2n}$ be the exponential kernel matrix with the dataset $Q \cup K$, and let s_λ be the statistical dimension of E (Definition 2.2), α be the row distortion of V (Definition C.2). There exists a quantum data structure (Algorithm 3) that preprocesses Q, K, V only through row queries to these matrices and with probability at least $1 - 1/\operatorname{poly}(n)$, for any $i \in [n]$, it outputs a vector $\widetilde{r}_i \in \mathbb{R}^d$ where*

$$\widetilde{r}_i = e_i^\top \widetilde{D}^{-1}\widetilde{A}\widetilde{V}.$$

If in addition, we have $\|D^{-1}\| < \frac{1}{\epsilon\|E\|+\lambda\sqrt{n}}$, then the approximations $\widetilde{D}, \widetilde{A}$ and $\widetilde{V}$ satisfy that

$$\begin{aligned}&\|(\widetilde{D}^{-1}(\widetilde{A} + \widetilde{A}^\top)/2 \cdot \widetilde{V}) - D^{-1}(A + A^\top)/2 \cdot V\|_F \\ &\leq \epsilon \cdot (\beta \cdot \|D^{-1}\|) \cdot (\|(A + A^\top)/2\|_F + \lambda\sqrt{n}) \cdot \|V\|_F,\end{aligned}$$

where $\beta = \frac{1}{1-(\epsilon\|E\|+\lambda\sqrt{n})\|D^{-1}\|}$. Moreover, the algorithm has the following runtime specification:

- *Preprocesses in $\widetilde{O}(\epsilon^{-1}n^{0.5}(s_\lambda^{2.5} + s_\lambda^{1.5}d + \alpha^{0.5}d) + d^\omega + s_\lambda^\omega + \epsilon^{-2}s_\lambda\alpha d)$ time;*
- *For any $i \in [n]$, it outputs $\widetilde{r}_i$ in $\widetilde{O}(s_\lambda^2 + s_\lambda d)$ time.*

Proof. By Theorem C.4, we know that with probability at least $1 - 1/\operatorname{poly}(n)$, the following bound holds:

$$\begin{aligned}\|\widetilde{D}^{-1}(\widetilde{A} + \widetilde{A}^\top)/2 \cdot S_V S_V^\top V\|_F &\leq \epsilon \cdot \|\widetilde{D}^{-1}(\widetilde{A} + \widetilde{A}^\top)/2\|_F \cdot \|V\|_F \\ &\leq \epsilon \cdot \|\widetilde{D}^{-1}\| \cdot \|(\widetilde{A} + \widetilde{A}^\top)/2\|_F \cdot \|V\|_F,\end{aligned}$$

where the second step is by $\|AB\|_F \leq \|A\| \cdot \|B\|_F$. By Theorem B.4, we know that

$$\|\widetilde{D} - D\| \leq \epsilon\|E\| + \lambda\sqrt{n},$$

note that as long as the error satisfies that $\|D^{-1}\| < \frac{1}{\epsilon\|E\|+\lambda\sqrt{n}}$, then by Lemma D.1, we obtain a bound on $\|\widetilde{D}^{-1}\|$:

$$\|\widetilde{D}^{-1}\| \leq \frac{\|D^{-1}\|}{1 - (\epsilon\|E\| + \lambda\sqrt{n})\|D^{-1}\|}.$$

Finally, by Corollary A.2, we have

$$\left\|\frac{\widetilde{A} + \widetilde{A}^\top}{2}\right\|_F \leq \left\|\frac{A + A^\top}{2}\right\|_F + \lambda\sqrt{n}.$$

For the runtime, it suffices to combine Corollary A.7, Theorem B.4 and Theorem C.4, and the only additional runtime term is the $\epsilon^{-2}s_\lambda\alpha d$, which is the time to form matrix $\widetilde{R}$ and $\widetilde{L}$. $\square$

LLM Usage Disclosure

LLMs were used only to polish language, such as grammar and wording. These models did not contribute to idea creation or writing, and the authors take full responsibility for this paper's content.