Support Basis: Fast Attention Beyond Bounded Entries

Maryam Aliakbarpour Rice University
Vladimir Braverman Johns Hopkins University
Junze Yin Rice University
Haochen Zhang Rice University

Abstract

Large language models (LLMs) have demonstrated remarkable performance across a wide range of tasks. However, the quadratic complexity of softmax attention remains a central bottleneck that limits their scalability. Alman and Song (NeurIPS 2023a; NeurIPS 2024a) proposed sub-quadratic time algorithms for attention inference and training, respectively, but they rely on the restrictive *bounded-entry assumption*. We show that this assumption rarely holds in practice, which significantly limits their applicability to modern LLMs.

In this paper, we introduce *support-basis decomposition*, a new technique for accurate and efficient attention inference and training *without* the bounded-entry assumption. We empirically show that the entries of the query and key matrices exhibit sub-Gaussian behavior. Leveraging this widely observed property, we perform exact computation on sparse components and polynomial approximation on dense components. Without relying on restrictive assumptions, we theoretically show that our algorithm achieves sub-quadratic runtime while matching the approximation error of prior work, and we empirically validate its computational efficiency and downstream task performance[1]. We further generalize our method to a multi-threshold setting that eliminates all distributional assumptions, providing the first theoretical justification for the empirical success of polynomial attention. Moreover, we show that softmax attention can be closely approximated by multiple polynomial attentions with significantly smaller ℓ_p error.

[1]Our code can be found at: https://github.com/yinj66/support_basis.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

1 INTRODUCTION

Large language models (LLMs) are powerful AI models for understanding and generating human-like text. Prominent examples include BERT (Devlin et al., 2019), OPT (Zhang et al., 2022b), GPT series (Brown et al., 2020; Radford et al., 2018, 2019; OpenAI, 2023), and Llama series (Touvron et al., 2023a,b; Dubey et al., 2024). These models are built upon the Transformer architecture (Vaswani et al., 2017), which utilizes the *softmax attention computation* as a core mechanism. Using this design, LLMs can handle a wide range of language tasks, like language modeling (Martin et al., 2020), sentiment analysis (Usama et al., 2020), creative writing (ChatGPT, 2022; OpenAI, 2023), and natural language translation (He et al., 2021).

The exact softmax attention computation enables models to emphasize the most influential parts of the input when generating outputs, rather than weighting all words uniformly. The attention weights, computed via a softmax function over the inputs, determine the relative importance assigned to each word. By selectively focusing on highly relevant inputs, attention layers allow models to handle longer sequences more effectively.

However, a crucial drawback of softmax attention is its quadratic time complexity in the input length n. Given query, key, and value matrices $Q, K, V \in \mathbb{R}^{n\times d}$ and entrywise exponential function exp, where $d \ll n$ is the hidden dimension, computing the (Q, K)-softmax attention matrix $A = \exp\left(QK^\top/d\right) \in \mathbb{R}^{n\times n}$ and the subsequent product AV are both computationally expensive, requiring $O\left(n^2 d\right)$ time[2]. This quadratic complexity makes it challenging for LLMs to process long input sequences efficiently. Since A contains n^2 entries, it is impossible to improve the asymptotic time complexity when explicitly computing all entries of A. Thus, prior works on attention optimization that aim

[2]In the standard attention literature (Vaswani et al., 2017), $A := \exp\left(QK^\top/\sqrt{d}\right)$. We use $\exp\left(QK^\top/d\right)$ solely for the simplicity in our theoretical analysis. We use the standard definition $\exp\left(QK^\top/\sqrt{d}\right)$ when running our experiments.

to improve quadratic complexity attempt to implicitly utilize A (Alman and Song, 2023a; Beltagy et al., 2020; Kitaev et al., 2020; Zaheer et al., 2020; Katharopoulos et al., 2020; Kacham et al., 2024; Wang et al., 2020; Gao et al., 2025a) to approximate it. Among these, Alman and Song (2023a) is the only work that provides a time-optimal algorithm for approximating softmax attention without reducing it to a simpler problem. Nevertheless, their algorithm relies on strong assumptions, which makes it very difficult to implement in modern transformers. To address this issue, we design an efficient algorithm for approximating attention via a novel mathematical technique, support-basis decomposition, eliminating the strong assumptions in Alman and Song (2023a). Additionally, we combine the polynomial method with sketching techniques to theoretically justify the competitive performance of polynomial attention (Kacham et al., 2024), where the entrywise exponential is replaced with an entrywise polynomial $p(x) = x^\beta$ for $\beta \geq 2$.

Organization In Section 1.1, we present the problem formulation and limitations of prior work. In Section 1.2, we address these challenges and go beyond.

1.1 Problem Setup and Prior Approaches

To justify the quadratic time complexity, we present the mathematical formulation of the exact attention computation. The (Q, K)-softmax-attention matrix $A = \exp\left(QK^\top/d\right) \in \mathbb{R}^{n\times n}$ captures relationships between each row of the query and the key by the inner product $A_{i,j} := \exp\left(\frac{1}{d}\langle Q_{i,*}, K_{j,*}\rangle\right)$, where $i, j \in [n] := \{1, 2, \ldots, n\}$, and $Q_{i,*}, K_{j,*} \in \mathbb{R}^d$ are the i-th and j-th rows of Q and K, respectively. The inner product $\langle Q_{i,*}, K_{j,*}\rangle = \|Q_{i,*}\|_2 \|K_{j,*}\|_2 \cos\theta$, computed in $O(d)$ time, quantifies the alignment of $Q_{i,*}$ and $K_{j,*}$, while the entrywise exponential function further accentuates these relationships by magnifying large similarities.

We further make each row of A form a probability distribution by dividing each entry by the corresponding row sum. Finally, contextual information is aggregated by multiplying the normalized attention matrix with the value matrix $V \in \mathbb{R}^{n\times d}$. The exact softmax attention computation problem is defined as follows:

Definition 1.1 (The exact attention computation)**.** *Given $Q, K, V \in \mathbb{R}^{n\times d}$, we let $A = \exp\left(QK^\top/d\right) \in \mathbb{R}^{n\times n}$ be the (Q, K)-softmax-attention matrix. For all $i, j \in [n]$, we let $\mathrm{diag} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ as $\mathrm{diag}(x)_{i,j} := x_i$ if $i = j$, and $\mathrm{diag}(x)_{i,j} := 0$ otherwise. Let $D := \mathrm{diag}(A\mathbf{1}_n) \in \mathbb{R}^{n\times n}$ and $\mathbf{1}_n \in \mathbb{R}^n$ be the all-ones vector. The exact attention computation $\mathrm{Att}(Q, K, V) \in \mathbb{R}^{n\times d}$ is defined as:*

$$\mathrm{Att}(Q, K, V) := D^{-1}AV.$$

To reduce the quadratic time complexity $O\left(n^2 d\right)$, Alman and Song (2023a) studies approximate attention computation to avoid fully constructing $A \in \mathbb{R}^{n\times n}$:

Definition 1.2 (The approximate attention computation)**.** *Let $\epsilon > 0$ be the accuracy parameter. For all matrix M, we let $\|M\|_\infty := \max_{i,j} |M_{i,j}|$. Given $Q, K, V \in \mathbb{R}^{n\times d}$, the goal of the approximate attention computation is to output a matrix $P \in \mathbb{R}^{n\times d}$ with:*

$$\left\|P - D^{-1}AV\right\|_\infty \leq \epsilon \cdot \|V\|_\infty.$$

Under the bounded entry assumption, Alman and Song (2023a) presents a time-optimal algorithm that approximates attention computation with small error:

- **Upper bound of Alman and Song (2023a)**: if the bounded entry assumption holds, that is with $B = o\left(\sqrt{\log n}\right)$, $\|Q\|_\infty, \|K\|_\infty, \|V\|_\infty \leq B$, then there exists a sub-quadratic time algorithm (Algorithm 2) that solves the approximate attention computation problem with $\epsilon = 1/\operatorname{poly}(n)$;
- **Lower bound of Alman and Song (2023a)**: if the bounded entry assumption fails, there does not exist any truly sub-quadratic time algorithm approximating attention computation under the Strong Exponential Time Hypothesis (SETH, Definition A.6) for $\epsilon = 1/\operatorname{poly}(n)$.

Building upon Alman and Song (2023a), Alman and Song (2024b) further shows that the gradient of the attention loss $\min_{X\in\mathbb{R}^{d\times d}} L(X)$:

$$\min_{X\in\mathbb{R}^{d\times d}} \|D(X)^{-1}\exp(X_\ell X X_\ell^\top)X_\ell W_V - E\|_F^2 \quad (1)$$

can be approximated in sub-quadratic time with the bounded entry assumption. $X_\ell \in \mathbb{R}^{n\times d}$ denotes the ℓ-th layer input, $D(X) = \mathrm{diag}\left(\exp(X_\ell X X_\ell^\top/d)\mathbf{1}_n\right)$, $X = W_Q W_K^\top$ is the variable, $E \in \mathbb{R}^{n\times d}$ denotes the desired output, and $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ are the weights for query, key, and value, respectively. This is equivalent to the attention definition (Definition 1.1), with the weights written explicitly, since attention backpropagation requires gradients with respect to the weights. When the bounded-entry assumption is violated, Alman and Song (2024b) shows that no truly sub-quadratic-time algorithm exists for approximating the gradient of the attention loss to accuracy $\epsilon = 1/\operatorname{poly}(n)$.

To obtain an accurate and efficient algorithm for approximating attention computation and its gradient (for attention inference and training, respectively), all entries in the matrices Q, K, V must be sufficiently small, i.e., $\max\{\|Q\|_\infty, \|K\|_\infty, \|V\|_\infty\} \leq o\left(\sqrt{\log n}\right)$. However, this bounded-entry assumption rarely holds in modern Transformer architectures (see Figure 1). Therefore, it is natural to ask:

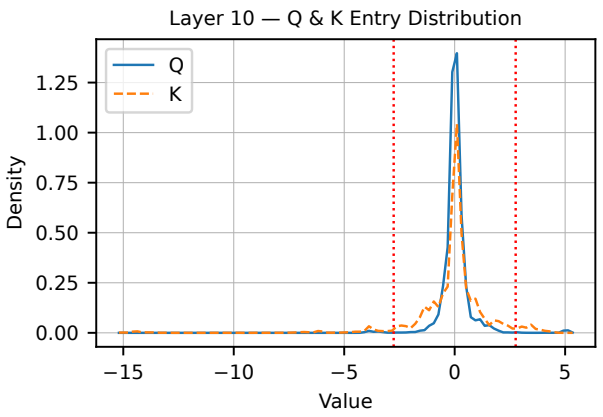


Figure 1: Distribution of entries in Q and K of Layer 10 in TinyLlama-1.1B (Zhang et al., 2024). The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

Can we accurately and efficiently approximate the attention computation with a practical assumption?

1.2 Our Results

Contribution 1—An Accurate, Efficient, and Implementable Algorithm We provide a positive answer to this question. Our "practical assumption" stems from the fact that the entries of the query, key, and value matrices resemble a sub-Gaussian distribution (Figure 1). These entries cluster near the mean, and the number of extreme entries—those whose absolute values exceed $o\left(\sqrt{\log n}\right)$—is small (see figures in Appendix L for more details beyond Figure 1: we provide empirical evidence of sub-Gaussian-like behavior in widely used modern Transformer architectures, such as TinyLlama-1.1B (Zhang et al., 2024), LLaDA-8B-Base (Nie et al., 2025), OPT-1.3B (Zhang et al., 2022b), and Phi-2 (Javaheripi et al., 2023), across *multiple* layers).

Under this practical assumption, the number of large entries is small, so we decompose the query Q into the sum of a sparse matrix, $Q^{(L)}$, whose rows contain large entries, and a dense matrix, $Q^{(s)}$, whose rows contain only small entries. We apply the same decomposition to the key K. We then perform the exact computation for the sparse components and use the polynomial method of Alman and Song (2023a) to approximate the small and dense component. We call this *single-threshold support basis decomposition*, where "single-threshold" denotes $T = o\left(\sqrt{\log n}\right)$ and "support basis" (Definition B.5) refers to the disjoint matrices $Q^{(s)}\left(K^{(s)}\right)^\top$ and $Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top$.

Theorem 1.3 (Informal version of Theorem D.4)**.** *Let $Q, K, V \in \mathbb{R}^{n\times d}$ be the query, key, and value matrices. Suppose entries of Q, K are independent and sub-Gaussian with variance proxies σ_Q^2 and σ_K^2, respectively. Let $\epsilon, \delta \in (0, 0.1)$ respectively be the accuracy parameter and failure probability. Then, with probability at least $1-\delta$, there exists a sub-quadratic time algorithm (Algorithm 1) that outputs $P \in \mathbb{R}^{n\times d}$ satisfying*

$$\left\|P - D^{-1}AV\right\|_\infty \leq \epsilon \cdot \|V\|_\infty,$$

where $A = \exp(QK^\top/d)$ and $D = \mathrm{diag}(A \cdot \mathbf{1}_n)$.

Using our single-threshold support basis, we can further approximate the gradient of the attention loss in sub-quadratic time if the entries of Q and K are sub-Gaussian. Unlike Alman and Song (2024b), our result enable efficient backward propagation without the bounded entry assumption (see Section 3.2 for detail).

Our empirical results on the distributions of the Q and K entries across multiple transformer models in multiple layers (e.g., Figure 1) imply that our sub-quadratic-time randomized algorithm can be applied to all of these models throughout the entire network. Since we show that both the forward and backward propagation of attention can be approximated in sub-quadratic time whenever sub-Gaussianity holds, we naturally extend our efficient algorithm to *multi-layer attention*, aligning it more closely with modern Transformer architectures.

However, it is also natural to ask:

Can we generalize our support basis to any arbitrary $Q, K \in \mathbb{R}^{n\times d}$ without any assumption?

Contribution 2—Efficient Attention Approximation With No Assumption We provide a positive answer to this question. Without any assumptions, entries of Q and K may not cluster near the mean, so we cannot infer that the number of large entries is small. Thus, we develop *multiple-threshold support basis* to approximate the attention computation in sub-quadratic time. However, due to the **lower bound** of Alman and Song (2023a), this unavoidably leads to a weaker accuracy guarantee $\epsilon > 1/\operatorname{poly}(n)$ in ℓ_∞ norm.

Theorem 1.4 (Informal version of Theorem I.5)**.** *Let $Q, K, V \in \mathbb{R}^{n\times d}$ be the query, key, and value matrices. Let $\epsilon, \delta \in (0, 0.1)$ respectively be the accuracy parameter and failure probability. Let $B = \max\{\|Q\|_\infty, \|K\|_\infty\}$ and $b = \min_{i\in[n], j\in[d]}\{|Q_{i,j}|, |K_{i,j}|\}$. Then, with probability $1-\delta$, the approximate attention computation (Definition 1.2) can be solved in sub-quadratic time by outputting $P \in \mathbb{R}^{n\times d}$ that satisfies*

$$\left\|P - D^{-1}AV\right\|_\infty \lesssim \epsilon \exp(3B^2) \cdot \|V\|_\infty .$$

Beyond the contribution in attention approximation, our result also provides a theoretical justification for why polynomial attention performs well in practice.

While Kacham et al. (2024) proposes a fast algorithm to approximate a polynomial attention and empirically demonstrates its effectiveness on downstream tasks, a theoretical explanation for this success remains lacking.

Why does the polynomial attention have comparable performance with the softmax attention?

Contribution 3—A Theoretical Justification for the Empirical Performance of Polynomial Attention and Beyond In our analysis of Theorem 1.4, we bridge this gap by showing that softmax attention is ϵ_1-close to a polynomial attention, and this polynomial attention can be approximated via the sketching methods of Kacham et al. (2024) with accuracy ϵ_2. It implies the resulting approximation is $(\epsilon_1+\epsilon_2)$-close to softmax attention, which explains why Kacham et al. (2024) obtains experimental results for polynomial attention that are comparable to those of softmax attention.

Moreover, we note that the attention approximation method of Alman and Song (2023a) does not perform well when the entries of Q and K are far apart. This motivates us to design a *multiple-threshold support basis*, which partitions Q and K into several disjoint components. For the partitions with larger entries, instead of performing exact computation as in the *single-threshold support basis*, whose running time becomes large without the sub-Gaussian assumption, we approximate them by a sum of polynomial attentions. It supports the empirical results of Kacham et al. (2024) and goes further: we can significantly reduce the ℓ_p error when approximating softmax attention using multiple polynomial attentions compared to a single polynomial attention, although the ℓ_∞ error **lower bound** of Alman and Song (2023a) remains impossible to bypass. Thus, combining multiple polynomial attentions better captures the behavior of softmax attention than a single polynomial attention in Kacham et al. (2024).

Contribution 4—Experimental Results Besides empirically validating our assumption that the entries of query and key matrices resemble sub-Gaussian distributions, we also present runtime experiments showing that our method is faster than exact attention computation and achieves significantly lower error than Alman and Song (2023a). Additionally, we evaluate downstream tasks with both Alman and Song (2023a) and our method across various benchmarks. Since the assumptions of Alman and Song (2023a) rarely hold in practice, it yields 0% accuracy, whereas our method aligns with practical settings and achieves performance comparable to exact attention (see Section 4).

Notation. Let $\mathbb{Z}_+$ be the set of positive integers. For all sets Y, we use $|Y|$ to denote its cardinality. We define the combination $\binom{n}{r} := \frac{n!}{(n-r)!r!}$. For all $i \in [d]$ and $x, y \in \mathbb{R}^d$, we define $x \circ y \in \mathbb{R}^d$ as $(x \circ y)_i := x_i \cdot y_i$; for all $A, B \in \mathbb{R}^{n\times d}$, for all $i \in [n]$ and $j \in [d]$, $A \circ B \in \mathbb{R}^{n\times d}$ is defined as $(A \circ B)_{i,j} := A_{i,j} \cdot B_{i,j}$. For all $p \in \mathbb{Z}_+$, we define the ℓ_p norm of x as $\|x\|_p := \left(\sum_{i\in[d]} |x_i|^p\right)^{1/p}$ and $A^{\circ p} \in \mathbb{R}^{n\times d}$ as $(A^{\circ p})_{i,j} := A^p_{i,j}$. We define $\mathrm{supp}(A) := \{(i,j) \in [n] \times [d] \mid A_{i,j} \neq 0\}$. $\mathbf{1}_{n\times d}$ is the $n \times d$ matrix whose entries are all 1's. $\Pr[\cdot]$ denotes the probability. The expectation of the discrete random variable $X \in \mathbb{R}$ is $\mathbb{E}[X] := \sum_i x_i \cdot \Pr[X = x_i]$.

Roadmap. In Section 2, we present the related work. In Section 3, we give an overview of techniques we use to prove our main results (Theorem 1.3 and Theorem 1.4). In Section 4, we present our experimental results.

2 RELATED WORK

Polynomial Methods for Attention Approximation As the attention approximation algorithm of Alman and Song (2023a) is time optimal, it has inspired numerous follow-up works (Alman and Song, 2024a,b, 2025; Chen et al., 2024; Liang et al., 2024c). They generalize the polynomial method from Alman and Song (2023a) to establish upper and lower bounds for various attention-approximation–related problems. Alman and Song (2024a) extends this method to study tensor attention inference, showing that the p-th order tensor attention can only be approximated in almost linear time under an even stronger assumption: $\max\{\|Q\|_\infty, \|K\|_\infty, \|V\|_\infty\} \leq o\left(\sqrt[p]{\log n}\right)$. Similarly, Alman and Song (2024b) proves that the gradient of attention can be approximated in almost linear time, and Liang et al. (2024c) shows the same for the gradient of tensor attention. Moreover, Alman and Song (2025); Chen et al. (2024) apply the method of Alman and Song (2023a) to Rotary Position Embedding (RoPE) attention (proposed by Su et al. (2024)) inference and training, respectively. All of these works rely on the strict bounded-entry assumption and contain no experimental results. Although our method is specifically built upon Alman and Song (2023a), it can be generalized to all these works to eliminate the bounded-entry assumption, since they all rely on similar techniques.

A recent work by Gupta et al. (2026) develops upon Alman and Song (2023a) by finding the relationships between the entry bound $B > 0$ and different hidden dimensions d. They show that when $d = O(1)$ and $B = \mathrm{poly}(n)$, the approximate attention computation problem admits a truly subquadratic-time algorithm with runtime $\widetilde{O}(n^{2-1/d} \cdot \mathrm{poly}\log(B/\varepsilon))$. In contrast, when $d = 2^{\Theta(\log^* n)}$, they establish a conditional lower bound under the SETH, proving that any algorithm computing the approximate attention

computation problem must take $n^{2-o(1)}$ time. Finally, in the regime where $d = \text{poly}(n)$, the standard algorithm with runtime $O(n^2 d)$ is conditionally optimal. Although Gupta et al. (2026) gives a more flexible choice of d, it is still unclear how we can design a fast algorithm approximating the attention computation problem if there are entries in Q, K, V larger than B.

Other Attention Optimization Works Other attention approximation works also rely on different types of assumptions that are incomparable to our work. Reformer (Kitaev et al., 2020) introduces Locality-Sensitive Hashing (LSH) Attention, which reduces the computational complexity from $O\left(n^2\right)$ to $O\left(n \log n\right)$ by assuming that, for each query vector, only a small subset of the nearest key vectors substantially contributes to the softmax output. Longformer (Beltagy et al., 2020) and Big Bird (Zaheer et al., 2020) build on the sparse attention matrix assumption. Linear Transformer (Katharopoulos et al., 2020) replaces the softmax operation with a kernel-based formulation, and Kacham et al. (2024) replaces the entrywise exponential with an entrywise polynomial. Linformer (Wang et al., 2020) assumes the self-attention matrix is low-rank.

The most similar work to our *multiple-threshold support basis* result is Liang et al. (2024a), which is also motivated by alleviating strict assumptions in prior works and develops a sub-quadratic time algorithm for attention approximation with a causal mask. However, their approach is limited to the case where the causal mask is a lower-triangular matrix in which all lower-triangular entries are equal to 1. Consequently, when applied with this mask, the attention matrix also becomes lower triangular. Liang et al. (2024a) decomposes the masked attention into a set of k "conv bases" with $k \leq n$ and analyzes each basis in $O(n \log n)$ time via Fast Fourier Transform to achieve sub-quadratic runtime. While this is similar in spirit to our approach, our task is significantly more challenging: we decompose a dense $n \times n$ attention matrix with no zero entries into a sum of disjoint components and analyze each using the polynomial method and sketching. Both works share the advantage of tuning the number of bases, but our method is more promising because our bucketing scheme generates at most a logarithmic number of bases. In contrast, the number of bases in Liang et al. (2024a) can be as large as n in the worst case, which yields no improvement in runtime.

3 TECHNIQUE OVERVIEW

Our theoretical proofs are deferred to the appendix. Below, we first discuss the origin and limitations of the bounded-entry assumption in attention optimization (Section 3.1). We then summarize the techniques used in our proofs to remove this bounded-entry assumption and support our main results. In Section 3.2, we show that a single-threshold support basis suffices to eliminate this assumption, proving Theorem 1.3. In Section 3.3, we introduce multiple-threshold support bases to prove Theorem 1.4.

3.1 Bounded Entry Assumption Limitations

In Lemma 3.4 of Alman and Song (2023a) (Lemma A.4), with $g = O\left(\max\left\{\frac{\log(1/\epsilon)}{\log(\log(1/\epsilon)/B)}, B^2\right\}\right)$ being the degree of the Chebyshev polynomial, if the bounded entry assumption hold ($B = o\left(\sqrt{\log n}\right)$), then there exists an algorithm that returns $U_1, U_2 \in \mathbb{R}^{n \times r}$ such that the attention matrix can be low-rank approximated $A \approx U_1 U_2^\top$, with the rank $r = \binom{2(d+g)}{2g}$. The construction of the low-rank approximation is by using a degree-g entrywise Chebyshev polynomial $U_1 U_2^\top = p\left(QK^\top/d\right)$ (see details in Appendix A.1) to approximate the attention matrix $\exp\left(QK^\top/d\right)$ so that computing $U_1\left(U_2^\top V\right)$ ($O\left(nrd\right)$ time) takes less time than AV ($O\left(n^2 d\right)$ time) if $r < n$. If the bounded entry assumption is not satisfied, i.e. $B \geq \Omega(\sqrt{\log n})$, then $g \geq c_1 \log n$ for some $c_1 > 1$, regardless of ϵ. Therefore, with $d = c_2 \log n$ for some $c_2 > 1$, we get

$$\binom{2(d+g)}{2g} \geq \left(\frac{d+g}{d}\right)^{2d} \geq 2^{\Omega(d)} \geq 2^{\Omega(\log n)} \geq n^{\Omega(1)}.$$

It implies that the attention approximation using $U_1, U_2 \in \mathbb{R}^{n \times n^{\Omega(1)}}$ requires more than n^2 time, regardless of the error ϵ. This is a significant drawback: if $B \geq \Omega(\sqrt{\log n})$, then the Chebyshev polynomial cannot yield a sub-quadratic time algorithm. If we force $g < o\left(\log n\right)$ to ensure $r < n$, the error guarantee breaks. Thus, directly using the sub-quadratic algorithm from Alman and Song (2023a) when $B \geq \Omega(\sqrt{\log n})$ may fail to provide any meaningful accuracy guarantee.

3.2 Single-Threshold Support Basis

To eliminate the bounded entry assumption, the most straightforward approach is to scan all entries of $Q, K \in \mathbb{R}^{n \times d}$ and move those larger than the threshold $T = o\left(\sqrt{\log n}\right)$ into $Q^{(L)}$ and $K^{(L)}$, respectively. The matrices containing the remaining smaller entries are denoted $Q^{(s)}$ and $K^{(s)}$. We denote this process as $\textsc{Split}(Q, T)$ and $\textsc{Split}(K, T)$. We perform exact computation for the terms related to $Q^{(L)}$ and $K^{(L)}$, and use the polynomial method (Alman and Song, 2023a) to approximate $Q^{(s)}\left(K^{(s)}\right)^\top$. However, we notice that since exp is applied entry-wisely, $(\exp\left(A + B\right)) \cdot V = (\exp\left(A\right) \circ \exp\left(B\right)) \cdot V$, which cannot be further simplified as multiplication $\cdot$ is not dis-

tributive over Hadamard product $\circ$. Therefore, we cannot divide and conquer it using this naive approach.

Divide: Design a Valid Support Basis Our technique is to design a novel decomposition method that breaks down $QK^\top/d$ into a sum of *disjoint matrices*—For a set of matrices $\{A_i\}_{i\in[m]} \subset \mathbb{R}^{n\times d}$ where $m \in \mathbb{Z}_+$, A_i is said to be a disjoint matrix if for all $(j,k) \in [n]\times[d]$, for all $i' \neq i$, if $(A_i)_{j,k} \neq 0$, then $(A_{i'})_{j,k} = 0$ (Definition B.1). With two disjoint matrices $A^{(s)}, A^{(L)}$ satisfying $QK^\top = Q^{(L)}K^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(s)}\right)^\top = A^{(s)} + A^{(L)}$, we define $\{A^{(s)}, A^{(L)}\}$ as a *support basis* of $QK^\top$. In our work, we construct $A^{(s)}$ as a dense matrix, and all of its entries are bounded by $o\left(\sqrt{\log n}\right)$; we construct $A^{(L)}$ as a sparse matrix, and its entries can be large. Therefore, $A^{(s)}$ is suitable for approximation using the polynomial method, whereas $A^{(L)}$ is suitable to be computed exactly since its sparsity ensures the fast matrix multiplication. By Fact B.2, this decomposition has a desirable mathematical property, namely $\exp\left(QK^\top/d\right)$ equals:

$$\exp\left(A^{(s)}/d\right) + \exp\left(A^{(L)}/d\right) - \mathbf{1}_{n\times n}. \tag{2}$$

This additive form is essential for the divide-and-conquer strategy, as multiplication $\cdot$ is distributive over addition $+$. This can break the original attention computation problem $D^{-1}\exp\left(QK^\top/d\right)V$ into a sum of two simpler problems $D^{-1}\exp\left(A^{(s)}/d\right)V$ and $D^{-1}\left(\exp\left(A^{(L)}/d\right) - \mathbf{1}_{n\times n}\right)V$. Before presenting how we conquer each of them, we first present how to construct such a valid support basis $\{A^{(s)}, A^{(L)}\}$. All of the entries of $Q^{(L)}K^\top + Q^{(s)}\left(K^{(L)}\right)^\top$ are "potentially large", so they should be included in $A^{(L)}$. To further ensure that our current $A^{(L)}$ is disjoint from the remaining dense component $Q^{(s)}\left(K^{(s)}\right)^\top$, for all $(i,j) \in [n]\times[n]$ satisfying $\left(Q^{(L)}K^\top + Q^{(s)}\left(K^{(L)}\right)^\top\right)_{i,j} \neq 0$, we extract the corresponding entry $\left(Q^{(s)}\left(K^{(s)}\right)^\top\right)_{i,j}$ and add it into $\left(A^{(L)}\right)_{i,j}$. Regarding $A^{(s)}$, we define $\left(A^{(s)}\right)_{i,j}$ to be $\left(Q^{(s)}\left(K^{(s)}\right)^\top\right)_{i,j}$ if $\left(A^{(L)}\right)_{i,j} = 0$ and 0 otherwise. Then, this forms a valid support basis.

Conquer: Approximate the Attention in $A^{(L)}$ Sparsity Time Now, we have shown that we can construct a support basis $\{A^{(s)}, A^{(L)}\}$ to make attention computation be divided into the sum of two smaller problems: $D^{-1}\exp\left(A^{(s)}/d\right)V$ and $D^{-1}\left(\exp\left(A^{(L)}/d\right) - \mathbf{1}_{n\times n}\right)V$. Below, we justify the statement that if the number of large entries (those greater than $o\left(\sqrt{\log n}\right)$) of Q and K is upper bounded by $O\left(n^\alpha\right)$ with $\alpha \in (0,1)$, then the attention computation can be approximated in sub-quadratic time.

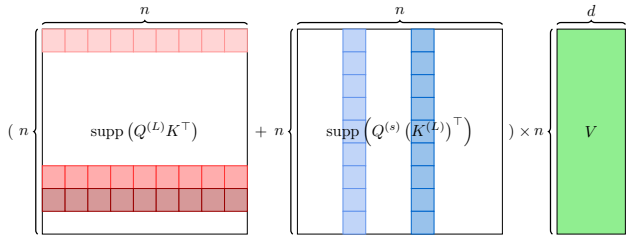


Figure 2: A visualization of $A^{(L)}V$. By the definition of $A^{(L)}$, $\text{supp}\left(A^{(L)}\right) = \text{supp}\left(Q^{(L)}K^\top + Q^{(s)}\left(K^{(L)}\right)^\top\right)$. Therefore, visualizing $A^{(L)}V$ is equivalent to visualize $\left(Q^{(L)}K^\top + Q^{(s)}\left(K^{(L)}\right)\right)V$. By definition, $Q^{(L)}, K^{(L)}$ are sparse matrices with $\left|\text{supp}\left(Q^{(L)}\right)\right| = \left|\text{supp}\left(K^{(L)}\right)\right| = O\left(n^\alpha\right)$ for $\alpha \in (0,1)$, so there are at most $O\left(n^\alpha\right)$ non-zero rows (red blocks) in $Q^{(L)}K^\top$ and $O\left(n^\alpha\right)$ non-zero columns (blue blocks) in $Q^{(s)}\left(K^{(L)}\right)^\top$.

To conquer each of these smaller problems, we efficiently approximate the former using polynomial method from Alman and Song (2023a) and exactly compute the latter. To ensure that our attention approximation algorithm runs in sub-quadratic time, it suffices to make the exact computation part sub-quadratic, since the polynomial method always takes $n^{1+o(1)}$ time. Recall that we define $A^{(L)} \in \mathbb{R}^{n\times n}$ as $\left(A^{(L)}\right)_{i,j} := \left(QK^\top\right)_{i,j}$ if $\left(Q^{(L)}K^\top + Q^{(s)}\left(K^{(L)}\right)^\top\right)_{i,j} \neq 0$ and 0 otherwise. Since $Q, K \in \mathbb{R}^{n\times d}$, computing an arbitrary entry of $QK^\top$ only takes d time. Thus, we need to know how many of the entries of $A^{(L)}$ are non-zero. As the number of large entries of Q and K is at most $O\left(n^\alpha\right)$ with $\alpha \in (0,1)$, we consider $Q^{(s)}$ and K to be dense and $Q^{(L)}, K^{(L)}$ to be sparse. Thus, the support of $A^{(L)}$ is represented as in Figure 2. We get $\text{supp}\left(\exp\left(A^{(L)}/d\right) - \mathbf{1}_{n\times n}\right) = \text{supp}\left(A^{(L)}\right)$ as $\exp(0) - 1 = 0$. When computing $\left(\exp\left(A^{(L)}/d\right) - \mathbf{1}_{n\times n}\right)V$, it suffices to consider:

- **Case 1**: if a matrix $A_1 \in \mathbb{R}^{n\times n}$ is sparse with n^α rows (red blocks in Figure 2) that are non-zero and the rest of the rows are all 0, then we only need to compute $n^\alpha d$ numbers of inner products when computing $A_1V \in \mathbb{R}^{n\times d}$. As each inner product takes $O(n)$ time, it takes $O(n^{1+\alpha}d)$ time in total to compute A_1V.
- **Case 2**: if a matrix $A_2 \in \mathbb{R}^{n\times n}$ is sparse with n^α columns (blue blocks in Figure 2) that are non-zero and the rest of the rows are all 0, computing $A_2V \in \mathbb{R}^{n\times d}$ requires nd numbers of inner product, but since each inner product takes only $O(n^\alpha)$ time, then in total, it still takes $O(n^{1+\alpha}d)$ time.

Combining both cases together, with d =

$O(\log n)$, it takes $O\left(n^{1+\alpha}\right)$ time to compute $\left(\exp\left(A^{(L)}/d\right) - \mathbf{1}_{n\times n}\right)V$. As $O\left(n^{1+\alpha}\right)$ is the sparsity of $A^{(L)}$ (Lemma B.9), we call it $A^{(L)}$ sparsity time.

Algorithm 1 We approximate the attention computation problem using the support basis $\{A^{(s)}, A^{(L)}\}$.

procedure ApproxAttention($Q \in \mathbb{R}^{n\times d}, K \in \mathbb{R}^{n\times d}, V \in \mathbb{R}^{n\times d}, n, d, \epsilon, T$)
  $Q^{(L)}, Q^{(s)} \in \mathbb{R}^{n\times d} \leftarrow$ Split(Q, T)
  $K^{(L)}, K^{(s)} \in \mathbb{R}^{n\times d} \leftarrow$ Split(K, T)
  Explicitly compute $A^{(L)}$.
  $U_1, U_2 \in \mathbb{R}^{n\times r} \leftarrow$ Polynomial($A^{(s)}, \epsilon$). ▷ To approximate $\exp\left(A^{(s)}/d\right)$.
  $d_1 \leftarrow U_1\left(U_2^\top \mathbf{1}_n\right) \in \mathbb{R}^n$.
  $d_2 \leftarrow \left(\exp\left(A^{(L)}/d\right) - \mathbf{1}_{n\times n}\right)\mathbf{1}_n \in \mathbb{R}^n$.
  $D^{-1} \leftarrow \mathrm{diag}\left(d_1 + d_2\right)^{-1} \in \mathbb{R}^{n\times n}$.
  $C_1 \leftarrow U_1\left(U_2^\top V\right) \in \mathbb{R}^{n\times d}$.
  $C_2 \leftarrow \left(\exp\left(A^{(L)}/d\right) - \mathbf{1}_{n\times n}\right)V \in \mathbb{R}^{n\times d}$.
  return $D^{-1}(C_1 + C_2)$.
end procedure

Constructing $D = \mathrm{diag}\left(\exp\left(QK^\top/d\right)\mathbf{1}_n\right)$ also takes $A^{(L)}$ sparsity time: because of the linearity property of $\mathrm{diag} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$, the time complexity is dominated by computing $\left(\exp\left(A^{(L)}/d\right) - \mathbf{1}_{n\times n}\right)\mathbf{1}_n$, which, similarly, takes $O\left(n^{1+\alpha}\right)$ time. Finally, as D^{-1} is diagonal, the attention computation problem can be approximated in $A^{(L)}$ sparsity time.

Sub-Gaussianity of Query and Key We have now justified the statement that if the number of large entries (those greater than $o\left(\sqrt{\log n}\right)$) of Q and K is upper bounded by $O\left(n^\alpha\right)$ with $\alpha \in (0,1)$, then the attention computation can be approximated in sub-quadratic time. What remains is to justify why this "if" condition holds. This is where our assumption—that the entries of the query and key matrices follow a sub-Gaussian distribution—becomes crucial. Suppose $Q, K \in \mathbb{R}^{n\times d}$ are random matrices with independent sub-Gaussian entries having variances σ_Q^2 and σ_K^2, respectively, we have $\Pr\left[|Q_{i,j}| \geq t\right] \leq 2\exp\left(-\frac{t^2}{\sigma_Q^2}\right)$ and $\Pr\left[|K_{i,j}| \geq t\right] \leq 2\exp\left(-\frac{t^2}{\sigma_K^2}\right)$. This implies that the expected number of large entries in $M^{(L)} \in \{Q^{(L)}, K^{(L)}\}$ is bounded as: $\mathbb{E}\left[\left|\mathrm{supp}\left(M^{(L)}\right)\right|\right] \leq 2nd \cdot \exp\left(-\frac{T^2}{\sigma_M^2}\right)$. Applying the multiplicative Chernoff bound, we obtain that with high probability, our "if" condition holds:

$$\Pr\left[\left|\mathrm{supp}\left(M^{(L)}\right)\right| > n^\alpha\right] \leq \exp\left(-\Omega\left(n^\alpha\right)\right).$$

Putting everything together, we conclude the proof sketch of Theorem 1.3 by Modus Ponens.

Generalize to Multi-Layer Attention Generalizing our method to multi-layer attention requires efficient algorithms for both inference and training. Inference can be achieved using Algorithm 1, while training involves efficiently approximating the gradient of the loss function (Eq. (1)). As noted in Alman and Song (2024b), the dominant term in the time complexity of computing $\frac{\mathrm{d}L}{\mathrm{d}X}$ still arises from the attention matrix $A \in \mathbb{R}^{n\times n}$. Thus, replacing A with the low-rank matrix $U_1U_2^\top$ like Alman and Song (2023a), Alman and Song (2024b) can approximate $\frac{\mathrm{d}L}{\mathrm{d}X}$ in sub-quadratic time.

On the other hand, our method expresses A as $\exp\left(A^{(s)}/d\right) + \exp\left(A^{(L)}/d\right) - \mathbf{1}_{n\times n}$ (Eq. (2)). Since the entries of $\exp\left(A^{(s)}/d\right)$ are small and satisfy the bounded entry assumption, we can approximate it by $U_1U_2^\top$ with little sacrifice in accuracy. However, although $\mathcal{B} = \exp\left(A^{(L)}/d\right) - \mathbf{1}_{n\times n}$ is sparse, its product with other dense matrices is dense, and thus further computation may still require quadratic time $\Omega\left(n^2\right)$. To address this issue, we use the approximate SVD from Clarkson and Woodruff (2017), allowing us to approximate the best rank-k low-rank approximation of $\mathcal{B}$ in $A^{(L)}$ sparsity time (Theorem E.3). Choosing $k = n^{o(1)}$, we approximate $\frac{\mathrm{d}L}{\mathrm{d}X}$ in $A^{(L)}$ sparsity time, which is also sub-quadratic by sub-Gaussianity, thereby generalizing our method to multi-layer attention.

3.3 Multiple-Threshold Support Basis

The single-threshold support basis works well when Q and K only have a small number of "large" entries. However, if we do *not* make any distributional assumptions, we cannot bound the number of large entries in Q and K with high probability. Thus, the exact computation may become too expensive.

Divide: Design a Multiple-Threshold Support Basis To address it, we develop the *multiple-threshold support basis* to decompose Q and K into several disjoint components by a sequence of thresholds $0 < T_1 < T_2 < \cdots < T_m$. These thresholds partition the range of entry magnitudes into m disjoint intervals. For example, entries within $(T_{\ell-1}, T_\ell]$ are assigned to the ℓ-th "bucket". We write $Q = Q^{(T_1)} + Q^{(T_2)} + \cdots + Q^{(T_m)}$ and $K = K^{(T_1)} + K^{(T_2)} + \cdots + K^{(T_m)}$. To ensure the disjointness of each $A^{(T_\ell, T_{\ell'})} := Q^{(T_\ell)}\left(K^{(T_{\ell'})}\right)^\top$, if we find a large entry from Q or K in the interval $(T_{\ell-1}, T_\ell]$, we take the entire row containing this entry and include it in $Q^{(T_\ell)}$ or $K^{(T_\ell)}$. We then expand the product $QK^\top$ as $QK^\top = \sum_{\ell=1}^m \sum_{\ell'=1}^m A^{(T_\ell, T_{\ell'})}$. Since there are m^2 such matrices $A^{(T_\ell, T_{\ell'})}$, we choose T_ℓ to grow exponentially with respect to ℓ to keep m^2 small. Thus, we only need to handle a small number of $A^{(T_\ell, T_{\ell'})}$ to achieve the desired running time. Due to its disjointness, the support basis $\{A^{(T_l, T_{l'})}\}_{T_l, T_{l'} \in \{T_\ell\}_{\ell=1}^m}$ admits a desir-

able additive decomposition (by induction) analogous to Eq. (2) for the single-threshold support basis.

Conquer: Reduce Gaussian Kernels to Polynomial Kernels and Approximate Them by Sketching The largest entries in each of $Q^{(T_\ell)}(K^{(T_{\ell'})})^\top$ is at most $d \cdot T_\ell \cdot T_{\ell'}$. Therefore, to make each of them satisfy the bounded entry assumption, we need to take out a scalar from this matrix $Q^{(T_\ell)}(K^{(T_{\ell'})})^\top = \frac{T_\ell \cdot T_{\ell'}}{\log n} \cdot Q^{(\ell)}(K^{(\ell')})^\top$ so that the ℓ_∞ norm of $Q^{(\ell)}(K^{(\ell')})^\top$ is small. As exp is applied entry-wisely, we can get $\exp(Q^{(T_\ell)}(K^{(T_{\ell'})})^\top) = \exp(Q^{(\ell)}(K^{(\ell')})^\top)^{\circ \frac{T_\ell \cdot T_{\ell'}}{\log n}}$, where the power $\frac{T_\ell \cdot T_{\ell'}}{\log n}$ is also applied entry-wisely on the matrix $\exp(Q^{(\ell)}(K^{(\ell')})^\top)$. Then, we apply the polynomial method (Alman and Song, 2023a) to approximate $\exp\left(Q^{(\ell)}(K^{(\ell')})^\top\right) \approx U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top$ in almost linear time. We can finally approximate the polynomial attention $D^{-1}\left(U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top\right)^{\circ \frac{T_\ell \cdot T_{\ell'}}{\log n}} V$ in linear time in n via oblivious sketching (Kacham et al., 2024; Ahle et al., 2020), which provides a randomized low-dimensional embedding so that $\left\langle \left(U_1^{(\ell,\ell')}\right)_{i,*}, \left(U_2^{(\ell,\ell')}\right)_{j,*} \right\rangle^{\frac{T_\ell \cdot T_{\ell'}}{\log n}} \approx \left\langle \phi'\left(\left(U_1^{(\ell,\ell')}\right)_{i,*}\right), \phi'\left(\left(U_2^{(\ell,\ell')}\right)_{j,*}\right) \right\rangle$ without explicitly computing the high dimensional inner product.

Error Analysis The reason to use bucketing is to control the range of values over which we approximate $\exp(x)$. Without bucketing, we would need to approximate $\exp(x)$ over the entire range of entries in $QK^\top/d$, which may contain both very small and very large values. If this range is wide, then we must take out a large scalar, which leads to a very high-degree polynomial. This will greatly increase the error, as by the mean value theorem, we can show $\|\mathcal{F}^{\circ p} - \mathcal{G}^{\circ p}\|_\infty \leq p \cdot \beta^{p-1} \cdot \|\mathcal{F} - \mathcal{G}\|_\infty$, where $\|\mathcal{F} - \mathcal{G}\|_\infty$ is the error of the Chebyshev polynomial (Alman and Song, 2023a) and β is the largest entry in $\mathcal{F}$ and $\mathcal{G}$. Bucketing addresses this issue by splitting the entries into groups (buckets) according to their magnitude. In each bucket, the maximum absolute value B_{bucket} is much smaller than the global maximum B. Thus, we can approximate $\exp(x)$ over $[-B_{\text{bucket}}, B_{\text{bucket}}]$ using a much lower-degree polynomial, reducing all ℓ_p error.

In attention computation, bucketing corresponds to decomposing the (Q, K)-softmax attention matrix into a sum of disjoint components, each of which can be approximated by a polynomial attention. This yields

$$\exp\left(QK^\top/d\right) \approx \sum_{\text{buckets } b} p_b\left(U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top\right).$$

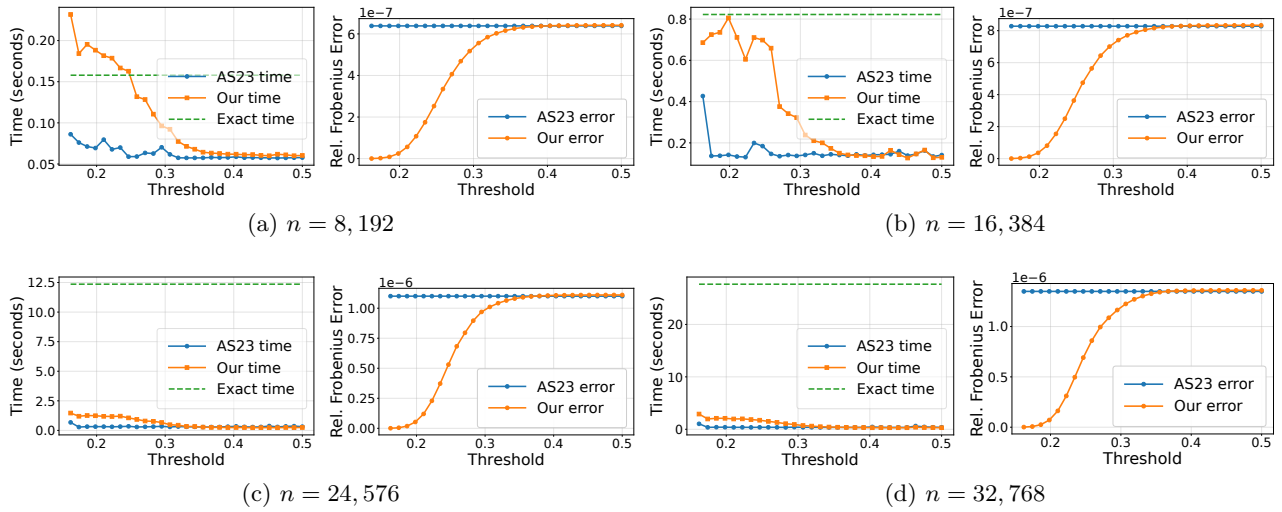
Each p_b is a polynomial tailored to the value range of its bucket. Because each polynomial is specialized to a narrower range, it matches the exponential function more closely within that range. Summing these accurate, range-specific approximations produces a result that is closer to the original softmax attention than using a single polynomial approximation over the full range. This concludes our proof sketch for Theorem 1.4.

4 EXPERIMENTAL RESULTS

Computational Efficiency We first compare the computational efficiency of the exact attention computation, the method of Alman and Song (2023a), and our proposed approach. We used an Apple M3 CPU to run this experiment. With $Q, K, V \in \mathbb{R}^{n \times d}$, we set $d = 64$ and $n \in \{8192, 16384, 24576, 32768\}$ consistent with real-world Transformer deployments. To satisfy our sub-Gaussian assumption, each entry of Q and K is drawn independently from a Gaussian distribution with mean 0 and standard deviation 0.1, i.e., for all $(i, j) \in [n] \times [d]$, $Q_{i,j} \sim \mathcal{N}(0, 0.1^2)$, $K_{i,j} \sim \mathcal{N}(0, 0.1^2)$.

Alman and Song (2023a) uses polynomial approximation to all entries, so it suffers a significant drop in accuracy due to the bounded-entry assumption. In contrast, our method allows for more flexible control of the accuracy–efficiency trade-off by partitioning large and small entries using a threshold $T > 0$. Small entries satisfies the bounded-entry assumption, so we apply polynomial approximation to gain efficiency with little loss in accuracy; for large entries, when the assumption is violated, we use exact computation. We evaluate our method with T ranging from 0.15 to 0.5. As T increases, our method performs more approximation and less exact computation. For all $n \in \{8192, 16384, 24576, 32768\}$, we can approximate the attention computation both efficiently and accurately by tuning T.

In addition, the approximation error ϵ is *not* driven by the sequence length n but instead is determined by the entry bound B in Q and K. Our experiment (right plot of each Figures 3a–3d) is consistent with the theoretical result in the Chebyshev approximation bound. When $\epsilon = 1/\text{poly}(n)$ and $d = O(\log n)$, the required polynomial degree satisfies $g = O\left(\max\left\{\frac{\log(1/\epsilon)}{\log(\log(1/\epsilon)/B)}, B^2\right\}\right)$ (Lemma 3.4 of Alman and Song (2023a)). Thus, when g and n are fixed, increasing B reduces the denominator $\log(\log(1/\epsilon)/B)$, which forces $\log(1/\epsilon)$ to shrink. Equivalently, ϵ must increase when B grows, which is exactly what we observe in our experiment (Figure 3).



(a) $n = 8,192$ (b) $n = 16,384$

(c) $n = 24,576$ (d) $n = 32,768$

Figure 3: For each of Figures 3a, 3b, 3c, and 3d, the left plot shows the change in running time with respect to the threshold T, while the right plot shows the change in error with respect to T. The blue curves correspond to Alman and Song (2023a), the orange curves correspond to our single-threshold support basis method (Algorithm 1), and the dotted green lines represent the exact attention computation (Definition 1.1).

Table 1: Performance of LLaDA-8B-Instruct with different attention approximation methods across various benchmarks. We evaluate the accuracy on these benchmarks. 4 and 6 are polynomial degrees. "Avg." is average.

	GSM8K	**MMLU**	**Hellaswag**	**ARC-easy**	**ARC-challenge**	**Avg.**
Exact computation	60.57%	62.56%	75.97%	91.62%	84.64%	75.47%
Our method-4	54.81%	61.17%	74.66%	89.90%	82.42%	72.59%
Our method-6	60.65%	62.62%	75.75%	91.58%	84.98%	75.11%
Alman and Song (2023a)-4	0%	0%	0%	0%	0%	0%
Alman and Song (2023a)-6	0%	0%	0%	0%	0%	0%

In contrast, the runtime heavily depends on n. Since our method reduces the attention computation from $O(n^2 d)$ to sub-quadratic time, the absolute runtime improvement becomes more pronounced as n increases. As shown in Figure 3, larger sequence lengths yield substantially larger speedups *without* introducing additional drawbacks in approximation error, making our method more suitable for larger transformer models with very long contexts, which aligns with the development of modern transformer architectures.

Performance on Downstream Tasks We evaluate the performance of LLaDA-8B-Instruct (Nie et al., 2025) on a range of standard benchmarks, including GSM8K (Cobbe et al., 2021), MMLU (Hendrycks et al., 2021), Hellaswag (Zellers et al., 2019), ARC-easy (Clark et al., 2018), and ARC-challenge (Clark et al., 2018). We perform the exact computation for the largest and smallest 10% of entries in Q, K and applies the polynomial approximation to the remaining 80%. In contrast, Alman and Song (2023a) approximates all entries of the attention matrix. Table 1 shows the results. Our method demonstrates a clear advantage over Alman and Song (2023a) in practical settings. With a degree-4 Chebyshev approximation, our approach incurs less than a 3% drop in accuracy compared to exact attention. With a degree-6 Chebyshev approximation, the performance is nearly indistinguishable from the original computation. By contrast, Alman and Song (2023a) fails to produce meaningful results, yielding zero accuracy across all benchmarks. The superiority of our method over Alman and Song (2023a) stems from two key factors. First, in multi-layer transformers, approximation errors in the attention computation accumulate across blocks; our hybrid strategy is substantially more robust to such error propagation. Second, large entries in Q and K are particularly sensitive to approximation error. By computing these entries exactly, our method avoids the large errors that Alman and Song (2023a) cannot, since it treats all entries uniformly.

Acknowledgment

Maryam Aliakbarpour is affiliated with the Ken Kennedy Institute at Rice University. Vladimir Braverman is supported in part by NSF Grant CNS 2528780.

References

Amol Aggarwal and Josh Alman. Optimal-degree polynomial approximations for exponentials and gaussian kernel density estimation. In *Proceedings of the 37th Computational Complexity Conference*, 2022.

Thomas D Ahle, Michael Kapralov, Jakob BT Knudsen, Rasmus Pagh, Ameya Velingker, David P Woodruff, and Amir Zandieh. Oblivious sketching of high-degree polynomial kernels. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 141–160. SIAM, 2020.

Maryam Aliakbarpour, Amartya Shankha Biswas, Kavya Ravichandran, and Ronitt Rubinfeld. Testing tail weight of a distribution via hazard rate. In *International Conference on Algorithmic Learning Theory*, pages 34–81. PMLR, 2023.

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

Josh Alman and Zhao Song. Fast attention requires bounded entries, 2023b. https://www.youtube.com/watch?v=S2C57hMO_8U&list=PLgKuh-lKre10-e2TWPCB0JdacFEnHHS31&index=2.

Josh Alman and Zhao Song. How to capture higher-order correlations? generalizing matrix softmax attention to kronecker computation. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=v0zNCwwkaV.

Josh Alman and Zhao Song. The fine-grained complexity of gradient computation for training large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=up4tWnwRol.

Josh Alman and Zhao Song. Fast roPE attention: Combining the polynomial method and fast fourier transform, 2025. URL https://openreview.net/forum?id=AozPzKE0oc.

Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D Smith, and Patrick White. Testing that distributions are close. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 259–269. IEEE, 2000.

Tugkan Batu, Ravi Kumar, and Ronitt Rubinfeld. Sublinear algorithms for testing monotone and unimodal distributions. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 381–390, 2004.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Lucien Birgé. On the risk of histograms for estimating decreasing densities. *The Annals of Statistics*, pages 1013–1022, 1987.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Clément L Canonne, Themis Gouleakis, and Ronitt Rubinfeld. Sampling correctors. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 93–102, 2016.

ChatGPT. Optimizing language models for dialogue. *OpenAI Blog*, November 2022. URL https://openai.com/blog/chatgpt/.

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)*, 2025a. URL https://openreview.net/forum?id=wso1gABiPZ.

Yifang Chen, Jiayan Huo, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Fast gradient computation for rope attention in almost linear time. *arXiv preprint arXiv:2412.17316*, 2024.

Yifang Chen, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. The computational limits of state-space models and mamba via the lens of circuit complexity. In *The Second Conference on Parsimony and Learning (Proceedings Track)*, 2025b. URL https://openreview.net/forum?id=bImlLT3r62.

Yifang Chen, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Fundamental limits of visual autoregressive transformers: Universal approximation abilities. In *Forty-second International Conference on Machine Learning*, 2025c.

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*, 2025d.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems, 2021. *URL https://arxiv.org/abs/2110.14168*, 9, 2021.

Yichuan Deng, Zhao Song, Shenghao Xie, and Chiwun Yang. Unmasking transformers: A theoretical approach to data recovery via attention weights. *arXiv preprint arXiv:2310.12462*, 2023a.

Yichuan Deng, Zhao Song, and Junze Yin. Faster robust tensor power method for arbitrary order. *arXiv preprint arXiv:2306.00406*, 2023b.

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, volume 1 (long and short papers)*, pages 4171–4186, 2019.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yeqi Gao, Zhao Song, Weixin Wang, and Junze Yin. A fast optimization view: Reformulating single layer attention in LLM based on tensor and SVM trick, and solving it in matrix multiplication time. In *The 41st Conference on Uncertainty in Artificial Intelligence*, 2025a. URL https://openreview.net/forum?id=qxRQ9f9zMv.

Yeqi Gao, Zhao Song, and Junze Yin. An iterative algorithm for rescaled hyperbolic functions regression. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025b. URL https://openreview.net/forum?id=xJU2GjcC1U.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36:19622–19635, 2023.

Yuzhou Gu, Zhao Song, Junze Yin, and Lichen Zhang. Low rank matrix completion via robust alternating minimization in nearly linear time. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=N0gT4A0jNV.

Yuzhou Gu, Zhao Song, and Junze Yin. Binary hypothesis testing for softmax models and leverage score models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=eKXicUVKCs.

Shreya Gupta, Boyang Huang, Barna Saha, Yinzhan Xu, and Christopher Ye. Subquadratic algorithms and hardness for attention with any temperature. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=PSaJZktut7.

Weihua He, Yongyun Wu, and Xiaohua Li. Attention mechanism for neural machine translation: A survey. In *2021 IEEE 5th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, volume 5, pages 1485–1489. IEEE, 2021.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1(3):3, 2023.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

Praneeth Kacham, Vahab Mirrokni, and Peilin Zhong. Polysketchformer: fast transformers via sketching polynomial kernels. In *International Conference on Machine Learning (ICML)*. PMLR, 2024.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

Yekun Ke, Xiaoyu Li, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, and Zhao Song. On computational limits and provably efficient criteria of visual autoregressive models: A fine-grained complexity analysis. *arXiv preprint arXiv:2501.04377*, 2025.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgNKkHtvB.

Zhihang Li, Zhao Song, Zifan Wang, and Junze Yin. Local convergence of approximate newton method for two layer nonlinear regression. *arXiv preprint arXiv:2311.15390*, 2023.

Zhihang Li, Zhizhou Sha, Zhao Song, and Mingda Wan. Attention scheme inspired softmax regression. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2025. URL https://openreview.net/forum?id=po92bv6yRD.

Jiehao Liang, Somdeb Sarkhel, Zhao Song, Chenbo Yin, Junze Yin, and Danyang Zhuo. A faster k-means++ algorithm. *arXiv preprint arXiv:2211.15118*, 2022.

Yingyu Liang, Heshan Liu, Zhenmei Shi, Zhao Song, Zhuoyan Xu, and Junze Yin. Conv-basis: A new paradigm for efficient attention inference and gradient computation in transformers. *arXiv preprint arXiv:2405.05219*, 2024a.

Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Yufa Zhou. Multi-layer transformers gradient can be approximated in almost linear time. *arXiv preprint arXiv:2408.13233*, 2024b.

Yingyu Liang, Zhenmei Shi, Zhao Song, and Yufa Zhou. Tensor attention training: Provably efficient learning of higher-order transformers. *arXiv preprint arXiv:2405.16411*, 2024c.

Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Mingda Wan. Hofar: High-order augmentation of flow autoregressive transformers. *arXiv preprint arXiv:2503.08032*, 2025.

Zhexiao Lin, Yuanyuan Li, Neeraj Sarna, Yuanyuan Gao, and Michael von Gablenz. Domain-shift-aware conformal prediction for large language models. *arXiv preprint arXiv:2510.05566*, 2025.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models. *Advances in Neural Information Processing Systems*, 37:122154–122184, 2024.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suarez, Yoann Dupont, Laurent Romary, Eric Villemonte de La Clergerie, Djame Seddah, and Benoit Sagot. Camembert: a tasty french language model. In *The 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

OpenAI. Gpt-4 technical report, 2023.

Feng Peiyuan, Yichen He, Guanhua Huang, Yuan Lin, Hanchong Zhang, Yuchen Zhang, and Hang Li. Agile: A novel reinforcement learning framework of llm agents. *Advances in Neural Information Processing Systems*, 37:5244–5284, 2024.

Eric Price, Zhao Song, and David P. Woodruff. Fast regression with an ℓ_∞ guarantee. In *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, 2017.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. ., 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Zhao Song, Weixin Wang, and Junze Yin. A unified scheme of resnet and softmax. *arXiv preprint arXiv:2309.13482*, 2023a.

Zhao Song, Mingquan Ye, Junze Yin, and Lichen Zhang. A nearly-optimal bound for fast regression with ℓ_∞ guarantee. In *International Conference on Machine Learning (ICML)*, pages 32463–32482. PMLR, 2023b.

Zhao Song, Junze Yin, and Ruizhe Zhang. Revisiting quantum algorithms for linear regressions: Quadratic speedups without data-dependent parameters. *arXiv preprint arXiv:2311.14823*, 2023c.

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

Zhao Song, Weixin Wang, Chenbo Yin, and Junze Yin. Fast and efficient matching algorithm with deadline instances. In *The Second Conference on Parsimony and Learning (Proceedings Track)*, 2025a. URL https://openreview.net/forum?id=TIneXGrWZt.

Zhao Song, Mingquan Ye, Junze Yin, and Lichen Zhang. Efficient alternating minimization with applications to weighted low rank approximation. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=rvhu4V7yrX.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

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*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Mohd Usama, Belal Ahmad, Enmin Song, M Shamim Hossain, Mubarak Alrashoud, and Ghulam Muhammad. Attention-based sentiment analysis using convolutional and recurrent neural network. *Future Generation Computer Systems*, 113:571–578, 2020.

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.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Xiangwen Wang, Jie Peng, Kaidi Xu, Huaxiu Yao, and Tianlong Chen. Reinforcement learning-driven llm agent for automated attacks on llms. In *Proceedings of the Fifth Workshop on Privacy in Natural Language Processing*, pages 170–177, 2024.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *The 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

Haochen Zhang, Zhiyun Peng, Junjie Tang, Ming Dong, Ke Wang, and Wenyuan Li. A multi-layer extreme learning machine refined by sparrow search algorithm and weighted mean filter for short-term multi-step wind speed forecasting. *Sustainable Energy Technologies and Assessments*, 50:101698, 2022a.

Haochen Zhang, Xi Chen, and Lin F Yang. Adaptive liquidity provision in uniswap v3 with deep reinforcement learning. *arXiv preprint arXiv:2309.10129*, 2023a.

Haochen Zhang, Junze Yin, Guanchu Wang, Zirui Liu, Lin Yang, Tianyi Zhang, Anshumali Shrivastava, and Vladimir Braverman. Breaking the frozen subspace: Importance sampling for low-rank optimization in LLM pretraining. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=ZdmmOAN4h3.

Haochen Zhang, Tianyi Zhang, Junze Yin, Oren Gal, Anshumali Shrivastava, and Vladimir Braverman. CoVE: Compressed vocabulary expansion makes better LLM-based recommender systems. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12575–12591, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.651. URL https://aclanthology.org/2025.findings-acl.651/.

Mumin Zhang, Yuzhi Wang, Haochen Zhang, Zhiyun Peng, and Junjie Tang. A novel and robust wind speed prediction method based on spatial features of wind farm cluster. *Mathematics*, 11(3):499, 2023b.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022b.

Zhi Zhang, Chris Chow, Yasi Zhang, Yanchao Sun, Haochen Zhang, Eric Hanchen Jiang, Han Liu, Furong Huang, Yuchen Cui, and OSCAR HERNAN MADRID PADILLA. Statistical guarantees for lifelong reinforcement learning using PAC-bayesian theory. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025c. URL https://openreview.net/forum?id=v9XFxkTl7m.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: memory-efficient llm training by gradient low-rank projection. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Checklist

1. For all models and algorithms presented, check if you include:

   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]

(b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]

(c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]
Justification: We will release our source code if our paper is accepted.

2. For any theoretical claim, check if you include:

(a) Statements of the full set of assumptions of all theoretical results. [Yes]

(b) Complete proofs of all theoretical results. [Yes]

(c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [No]
Justification: We will release our code, data, and instructions to reproduce figures and tables if our paper is accepted.

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
Justification: We did not train any models. However, for our experiment in which we run inference, we have provided training details, such as hyperparameters, in our paper.

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
Justification: In our paper, we do not use error bars.

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. [Yes]

(b) The license information of the assets, if applicable. [Yes]

(c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

(d) Information about consent from data providers/curators. [Not Applicable]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Support Basis: Fast Attention Beyond Bounded Entries: Supplementary Materials

Contents

Roadmap. Our paper presents two main results. First, we introduce the use of a *single threshold support basis* to separate the large and small entries of the query and key matrices $Q, K \in \mathbb{R}^{n \times d}$. Under the sub-Gaussian assumption, this allows us to design a sub-quadratic time algorithm for approximating the attention computation, without requiring the bounded-entry condition $\|Q\|_\infty, \|K\|_\infty < o\left(\sqrt{\log n}\right)$ used in Alman and Song (2023a). We first use Appendix A to present the background of Alman and Song (2023a); Kacham et al. (2024). Then, we use Appendix B, Appendix C, and Appendix D to support our result.

Specifically, in Appendix B, we formally define the support basis and show how to split the large and small entries of the query and key matrices $Q, K \in \mathbb{R}^{n \times d}$ in the attention computation. In Appendix C, we use our theoretical result on the *single threshold support basis* to approximate the attention computation in $A^{(L)}$ sparsity time. In Appendix D, we introduce our sub-Gaussian assumption and show that the number of large entries in Q and K is bounded, so that with high probability, the running time of our attention approximation algorithm, namely the $A^{(L)}$ sparsity time, is sub-quadratic. In Appendix E, we show how we can generalize our algorithm to multi-layer attention.

Second, we propose the use of a *multiple thresholds support basis* to decompose $Q, K \in \mathbb{R}^{n \times d}$ into the sum of several matrices according to these thresholds. For each resulting component, we apply the polynomial approximation method from Alman and Song (2023a) and the sketching technique from Ahle et al. (2020); Kacham et al. (2024) to approximate and solve all subproblems in sub-quadratic time—without making any distributional assumptions, though at the cost of reduced accuracy. We use Appendix F, Appendix G, Appendix H, and Appendix I to support this result.

Specifically, in Appendix F, we generalize the theoretical results of the *single threshold support basis* to the setting with multiple thresholds and explain how to decompose $Q, K \in \mathbb{R}^{n \times d}$ accordingly. In Appendix G, we specify how to choose the thresholds to ensure that the number of decomposed components remains bounded. In Appendix H, we adapt the sketching techniques from Ahle et al. (2020); Kacham et al. (2024) to our setting. In Appendix I, we combine the sketching techniques with our *multiple thresholds support basis* to approximate the attention computation in sub-quadratic time, without assuming bounded entries or any distributional conditions.

Finally, in Appendix J, we present the basic mathematical facts used throughout the paper to support our proofs. In Appendix K, we present additional related works. In Appendix L, we show the entry distributions of the query and key matrices on different transformer architectures across different layers.

Notation. For all positive integer n, d, we denote $\mathbb{R}, \mathbb{R}^n, \mathbb{R}^{n \times d}$ as sets containing all real numbers, all n-dimensional vectors, and $n \times d$ matrices, whose entries are all in $\mathbb{R}$. We define $[n] := \{1, 2, \ldots, n\}$. For all set X, we use $|X|$ to denote its cardinality, namely the number of elements in this set. For all sets X and Y, we define $X \times Y := \{(x, y) \mid x \in X, y \in Y\}$.

For all $a \in \mathbb{R}$, we define $\lfloor a \rfloor$ as the largest integer satisfying $\lfloor a \rfloor \leq a$. For all $x, y \in \mathbb{R}^d$, their inner product is $\langle x, y \rangle := \sum_{i=1}^d x_i y_i$. With any arbitrary $i \in [d]$, the Hadamard product $\circ$ is a binary operation: $x \circ y \in \mathbb{R}^d$ is defined as $(x \circ y)_i := x_i \cdot y_i$. For all p being a positive integer or ∞, we define the ℓ_p norm of x as $\|x\|_p := \left(\sum_{i \in [d]} |x_i|^p\right)^{1/p}$. We let $\mathbf{1}_d$ and $\mathbf{0}_d$ respectively denote the d-dimensional vectors whose entries are all 1's and 0's. Similarly, we respectively define $\mathbf{1}_{n \times d}$ and $\mathbf{0}_{n \times d}$ as the $n \times d$ matrix whose entries are all 1's and 0's. We define $\mathrm{diag}(x) \in \mathbb{R}^{d \times d}$ to be the diagonal matrix with $\mathrm{diag}(x)_{i,i} = x_i$.

Let $A \in \mathbb{R}^{n \times d}$. We let $A_{i,j} \in \mathbb{R}$ be the (i, j)-th entry of A, $A_{i,*} \in \mathbb{R}^d$ be the i-th row, and $A_{*,j} \in \mathbb{R}^n$ be the j-th column. We let $A^\top \in \mathbb{R}^{d \times n}$ be the transpose. $\|A\|_F, \|A\|_\infty \in \mathbb{R}$ respectively are the Frobenius norm and ℓ_∞ norm, where $\|A\|_F := \sqrt{\sum_{i \in [n]} \sum_{j \in [d]} |A_{i,j}|^2}$ and $\|A\|_\infty := \max_{i \in [n], j \in [d]} |A_{i,j}|$. We define $\mathrm{supp}(A) := \{(i, j) \in [n] \times [d] \mid A_{i,j} \neq 0\}$. Let n_1, n_2, d_1, d_2, p be positive integers. We define $\exp(A) \in \mathbb{R}^{n \times d}$ and $A^{\circ p} \in \mathbb{R}^{n \times d}$ as the entry-wise exponential and power, namely for all $i \in [n]$ and $j \in [d]$, we have $\exp(A)_{i,j} := \exp(A_{i,j})$ and

$(A^{\circ p})_{i,j} := (A_{i,j})^p$. We define $\|A\|_p := \sqrt[p]{\sum_{i,j} |A_{i,j}|^p}$. Let $A \in \mathbb{R}^{n_1 \times d_1}$. We let $A^{\otimes p}$ be $\underbrace{A \otimes A \otimes \cdots \otimes A}_{p} \in \mathbb{R}^{n_1 \times d_1^p}$ be the row-wise Kronecker product.

For a differentiable function f, we use f' to denote its derivative. For a probability space $(\Omega, \mathcal{F}, \Pr)$, where Ω is the sample space, $\mathcal{F}$ is the σ-algebra of events, and $\Pr : \mathcal{F} \to [0,1]$ is the probability measure, we define the discrete random variable $X : \Omega \to \mathbb{R}$ so that the expectation of X is $\mathbb{E}[X] := \sum_i x_i \cdot \Pr[X = x_i]$, and for all $A \in \mathcal{F}$, we define the indicator function $\mathbb{I}[A] : \Omega \to \{0,1\}$ as $\mathbb{I}[A](\omega) := 1$ if $\omega \in A$ and $\mathbb{I}[A](\omega) := 0$ if $\omega \notin A$, for all $\omega \in \Omega$.

A PRELIMINARIES

In this section, we provide the theoretical foundation necessary to understand and contextualize our contributions in attention approximation. We give a summary of the prior work (Alman and Song, 2023a; Aggarwal and Alman, 2022), which established sub-quadratic time algorithms for attention approximation under strong boundedness assumptions. We also present the important mathematical properties from Kacham et al. (2024); Ahle et al. (2020).

In Appendix A.1, we give an overview of techniques in Alman and Song (2023a); Aggarwal and Alman (2022). In Appendix A.2, we present the background related to the sketching technique and polynomial attention (Kacham et al., 2024; Ahle et al., 2020).

A.1 Background: Technique Overview of the Polynomial Method

Accuracy-Efficiency trade-off. Before delving into our method, we first provide an overview of the techniques used in Aggarwal and Alman (2022); Alman and Song (2023a) to explain the origin of the bounded entry assumption. The attention approximation (Definition 1.2) can be computed in sub-quadratic time by replacing the entry-wise exponential function $\exp : \mathbb{R} \to \mathbb{R}$ in the softmax unit with a degree-g Chebyshev polynomial $p : \mathbb{R} \to \mathbb{R}$, thereby reducing the time complexity. This polynomial is also applied entry-wisely to $QK^\top/d \in \mathbb{R}^{n \times n}$, where each entry $\langle Q_{i,*}, K_{j,*} \rangle / d$ is the inner product between a query and a key vector. When this polynomial is expanded, it produces a collection of monomials that are products of certain entries from Q and K. These terms can be reorganized into a low-rank (rank-r) factorization of the form $U_1 U_2^\top$, where $U_1, U_2 \in \mathbb{R}^{n \times r}$ and $r = \binom{2d+2g}{2d}$. Similar as Alman and Song (2023b), we give the following example to approximate the (i,j)-th entry of $\exp\left(QK^\top/d\right)$:

$$
\begin{aligned}
& p\left(\langle Q_{i,*}, K_{j,*} \rangle / d\right) \\
= \; & p\left(\frac{1}{d} \sum_{\ell=1}^{d} Q_{i,\ell} \cdot K_{j,\ell}\right) \\
= \; & 3 Q_{i,1} K_{j,1} Q_{i,3} K_{j,3} - 4 Q_{i,7} K_{j,7} Q_{i,16}^3 K_{j,16}^3 + \ldots && (3) \\
= \; & \Big\langle \underbrace{\begin{bmatrix} 3 Q_{i,1} Q_{i,3} \\ -4 Q_{i,7} Q_{i,16}^3 \\ \vdots \end{bmatrix}}_{(U_1)_{i,*}}, \underbrace{\begin{bmatrix} K_{j,1} K_{j,3} \\ K_{j,7} K_{j,16}^3 \\ \vdots \end{bmatrix}}_{(U_2)_{j,*}} \Big\rangle && (4) \\
= \; & (U_1)_{i,*}^\top \cdot (U_2)_{j,*} . && (5)
\end{aligned}
$$

For all $B > 0$, to accurately approximate $\exp(x)$ on the range $[-B, B]$, the degree g of the polynomial must grow with B (see Lemma A.4 for details). This leads to an accuracy–efficiency trade-off: increasing g improves the approximation accuracy, but also increases the rank $r = \binom{2d+2g}{2d}$, thereby slowing down the computation.

Time complexity for constructing $U_1, U_2 \in \mathbb{R}^{n \times r}$ and approximating the attention computation. We now explain why constructing $U_1, U_2 \in \mathbb{R}^{n \times r}$ takes $O(nrg)$ time and using them to approximate the attention computation takes $O(nrd)$ time. As shown in Eq. (5), each degree-g Chebyshev polynomial may give one row of U_1, denoted $(U_1)_{i,*}$, and one row of U_2, denoted $(U_2)_{j,*}$. As $i, j \in [n]$, we need n numbers of Chebyshev polynomials

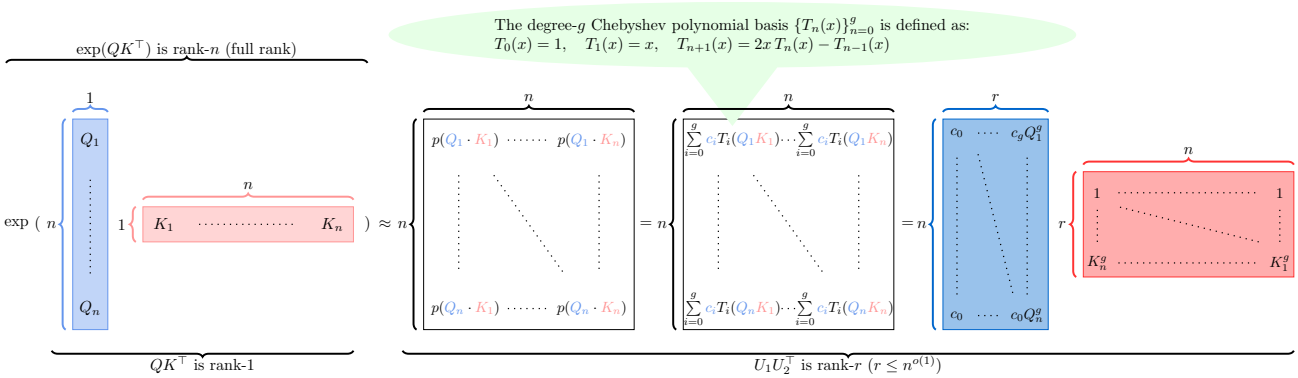


Figure 4: A visualization of the polynomial method. For simplicity, we set $d = 1$. The matrix $QK^\top$ has rank 1, but the entrywise exponential function exp may map it to a full-rank matrix. Alman and Song (2023a) shows that replacing exp with a degree-g Chebyshev polynomial p can produce a matrix of rank r.

to fully construct $U_1, U_2 \in \mathbb{R}^{n\times r}$. Each polynomial expansion contains r monomials, so each row vector $(U_1)_{i,*}$ or $(U_2)_{j,*}$ has r entries to be computed (see from Eq. (3) to Eq. (4) as an example). Additionally, if the polynomial is degree-g, computing the value of each entry of $(U_1)_{i,*}$ or $(U_2)_{j,*}$ involves multiplying up to $O(g)$ scalar factors, i.e., entries from Q or K (see entries in Eq. (4) as an example). Thus, each monomial evaluation costs $O(g)$ scalar multiplications. Since we compute n rows, each with r monomials, and each monomial costs $O(g)$ operations, constructing U_1 and U_2 requires $O(nrg)$ time. Now, we do not need to explicitly form the $n \times n$ attention matrix $A = \exp\left(QK^\top/d\right)$ but instead work with the much smaller factors U_1 and U_2 satisfying $A \approx U_1U_2^\top$. Finally, using the associativity of matrix multiplication, the attention computation (as in Definition 1.1)

$$\mathrm{diag}\left(\underbrace{U_1}_{n\times r}\left(\underbrace{U_2^\top}_{r\times n}\underbrace{\mathbf{1}_n}_{n\times 1}\right)\right)^{-1}\underbrace{U_1}_{n\times r}\left(\underbrace{U_2^\top}_{r\times n}\underbrace{V}_{n\times d}\right)$$

can be approximated in $O\left(nrd\right)$ time.

Limitations. To approximate the attention in almost linear time, we need to make $O(nrg + nrd) = O(n^{1+o(1)})$, which is equivalent to making $d, g, r \le n^{o(1)}$. However, keeping g and r small requires B (the maximum absolute entry of Q or K) to be small. The analysis in Alman and Song (2023a) shows that achieving both a fast runtime and a small approximation error requires $d = O(\log n)$ and $B = o\left(\sqrt{\log n}\right)$. This condition on B is known as the bounded entry assumption.

A.1.1 Background: Low Rank Approximation

In this section, we review concepts of matrix low-rank approximation. A key idea is to approximate the softmax attention matrix, which is dense and expensive to compute, using a low-rank decomposition constructed from polynomial expansions. This technique is central to the method proposed in Alman and Song (2023a), where the exponential function is approximated by a Chebyshev polynomial, and the resulting polynomial matrix is factorized into a product of two low-rank matrices. This allows for efficient computation of the attention matrix without explicitly forming it. We formally define the notion of an (ϵ, r)-approximation and present a key lemma from Alman and Song (2023a) that shows how to construct such a factorization efficiently.

Definition A.1. *Let $r \ge 1$ denote a positive integer. Let $\epsilon \in (0, 0.1)$ denote an accuracy parameter. Given a matrix $A \in \mathbb{R}_{\ge 0}^{n\times n}$, we say $\widetilde{A} \in \mathbb{R}_{\ge 0}^{n\times n}$ is an (ϵ, r)-approximation of A if*

- *$\widetilde{A} = U_1 \cdot U_2^\top$ for some matrices $U_1, U_2 \in \mathbb{R}^{n\times r}$ (i.e., $\widetilde{A}$ has rank at most r), and*
- *$|\widetilde{A}_{i,j} - A_{i,j}| \le \epsilon \cdot A_{i,j}$ for all $(i, j) \in [n]^2$.*

Lemma A.2 (Lemma 3.2 in Alman and Song (2023a))**.** *Let $M = XY^\top \in \mathbb{R}^{n\times n}$ denote a matrix with $X, Y \in \mathbb{R}^{n\times d}$. Let $P(x)$ denote a degree-g polynomial, and define $r = \binom{2(g+d)}{2g}$.*

There is an algorithm that runs in $O(nrg)$ time and, given as input the matrix X, Y, constructs matrices $U_1, U_2 \in \mathbb{R}^{n \times r}$ such that $P(M) = U_1 U_2^\top$. (Here, $P(M)$ denotes the entry-wise application of P to M.)

A.1.2 Background: Approximating the Attention Computation Assuming $d = O(\log n)$ and Bounded Entries

In this section, we describe the algorithm introduced in Alman and Song (2023a) for approximating the attention computation.

Algorithm 2 Algorithm 1 of Alman and Song (2023a). It takes $Q \in \mathbb{R}^{n \times d}, K \in \mathbb{R}^{n \times d}, V \in \mathbb{R}^{n \times d}$ as input and approximates the attention computation (Definition 1.2) under the assumptions $d = O(\log n)$, $\|Q\|_\infty \leq B$, $\|K\|_\infty \leq B$, and $\|V\|_\infty \leq B$, where $B = o\left(\sqrt{\log n}\right)$.

procedure PolyAttention($Q \in \mathbb{R}^{n \times d}, K \in \mathbb{R}^{n \times d}, V \in \mathbb{R}^{n \times d}, n, d, B, \epsilon$) ▷ Theorem A.5
▷ ϵ is the accuracy output
▷ $\|Q\|_\infty, \|K\|_\infty, \|V\|_\infty \leq B$
  $g \leftarrow O\left(\max\left\{\frac{\log(1/\epsilon)}{\log(\log(1/\epsilon)/B)}, B^2\right\}\right)$
  $r \leftarrow r = \binom{2(g+d)}{2g}$
  Construct $U_1, U_2 \in \mathbb{R}^{n \times r}$ via Lemma A.4 ▷ $O(nrg)$ time
  $\widetilde{w} \leftarrow U_1 \cdot \left(U_2^\top \mathbf{1}_n\right)$ ▷ $O(nr)$ time
  $\widetilde{D}^{-1} = \mathrm{diag}\left(\widetilde{w}^{-1}\right)$ ▷ $O(n)$ time
  Compute $U_2^\top V \in \mathbb{R}^{r \times d}$ ▷ Takes $\mathcal{T}_{\mathrm{mat}}(r, n, d)$ time
  Compute $U_1 \cdot \left(U_2^\top V\right)$ ▷ $\mathcal{T}_{\mathrm{mat}}(n, r, d)$ time
  $P \leftarrow \widetilde{D}^{-1} \cdot \left(U_1 \cdot \left(U_2^\top V\right)\right)$ ▷ $O(nd)$ time
  return P ▷ $P \in \mathbb{R}^{n \times d}$
end procedure

Lemma A.3 (Corollary 2.2 in Alman and Song (2023a))**.** *Let $B > 1$ and let $\epsilon \in (0, 0.1)$. There is a polynomial $P : \mathbb{R} \to \mathbb{R}$ of degree $g := \Theta\left(\max\left\{\frac{\log(1/\epsilon)}{\log(\log(1/\epsilon)/B)}, B\right\}\right)$ such that for all $x \in [-B, B]$, we have*

$$|P(x) - \exp(x)| < \epsilon \cdot \exp(x).$$

Lemma A.4 (An improved version of Lemma 3.4 in Alman and Song (2023a))**.** *Let $B = o\left(\sqrt{\log n}\right)$. Suppose $Q, K \in \mathbb{R}^{n \times d}$, with $\left\|QK^\top\right\|_\infty \leq dB^2$. Let $A := \exp\left(QK^\top/d\right) \in \mathbb{R}^{n \times n}$. For accuracy parameter $\epsilon \in (0, 1)$, there is a positive integer g bounded above by*

$$g = O\left(\max\left\{\frac{\log(1/\epsilon)}{\log\left(\log(1/\epsilon)/B\right)}, B^2\right\}\right),$$

and a positive integer r bounded above by

$$r \leq \binom{2(g+d)}{2g}$$

such that: There is a matrix $\widetilde{A} \in \mathbb{R}^{n \times n}$ that is an (ϵ, r)-approximation (Definition A.1) of $A \in \mathbb{R}^{n \times n}$. Furthermore, the matrices U_1 and U_2 defining $\widetilde{A}$ can be computed in $O(n \cdot r)$ time.

Proof. By the assumption that $\left\|QK^\top\right\|_\infty \leq dB^2$, we can get

$$\left\|QK^\top/d\right\|_\infty \leq B^2.$$

Thus, applying Lemma A.3 (with bound B^2 on its entries), there is a degree-g polynomial P such that the matrix $\widetilde{A} = P(M)$ is an (ϵ, r)-approximation to A. We can then compute U_1, U_2 using Lemma A.2, which gives the bound

$$r = \binom{2(g+d)}{2g}.$$

This completes the proof. □

A.1.3 Background: Main Results of Alman and Song (2023a)

In this section, we formally present the core theoretical results from Alman and Song (2023a), which establish both upper and lower bounds for the approximate attention computation problem. The upper bound shows that when the input matrices $Q, K, V \in \mathbb{R}^{n \times d}$ have entries bounded in ℓ_∞ norm by $B = o(\sqrt{\log n})$, and the hidden dimension satisfies $d = O(\log n)$, then the attention computation can be approximated to within error $\epsilon = 1/\mathrm{poly}(n)$ in almost linear time $n^{1+o(1)}$.

Theorem A.5 (Upper bound, Theorem 3.8 of Alman and Song (2023a)). *With $d = O(\log n), B = o(\sqrt{\log n}), \epsilon_a = 1/\mathrm{poly}(n)$), the approximate attention computation problem (see Definition 1.2) can be solved in time $\mathcal{T}_{\mathrm{mat}}(n, n^{o(1)}, d) = n^{1+o(1)}$.*

In contrast, the lower bound establishes a fundamental computational limitation: under the SETH, any algorithm that approximates attention within accuracy $\epsilon = 1/\mathrm{poly}(n)$ must take at least $n^{2-o(1)}$ time when $B = \Theta(\sqrt{\log n})$.

Definition A.6 (Strong Exponential Time Hypothesis (SETH), Hypothesis 4.1 in Alman and Song (2023a)). *For every $\epsilon > 0$ there is a positive integer $k \geq 3$ such that k-SAT on formulas with n variables cannot be solved in $O(2^{(1-\epsilon)n})$ time, even by a randomized algorithm.*

Theorem A.7 (Lower bound, Theorem 4.6 of Alman and Song (2023a)). *Assuming SETH, for every sufficiently small $q > 0$, there are constants $C > 0$ and $C_\alpha > 0$ and $C_\beta > 1$ such that Approximate Attention Computation (Definition 1.2) for parameters $d = O(\log n)$, $B = C_\beta \sqrt{\log n}$, and $\epsilon_a = n^{-C_\alpha}$ requires $\Omega(n^{2-q})$ time.*

A.2 Background: Sketching and Polynomial Attention

Below, we define a version of polynomial attention from Kacham et al. (2024).

Definition A.8 (Polynomial Attention (Kacham et al., 2024)). *Let $g : \mathbb{R} \to \mathbb{R}$ be defined as $g(z) = z^\beta$ for $\beta \geq 2$, where $g(W)_{i,j} = g(W_{i,j})$ if W is a matrix and $g(x)_i = g(x_i)$ if x is a vector. Given the input sequence $X \in \mathbb{R}^{n \times d}$ and the query, key, and value weights $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$, the polynomial attention computation is defined as:*

$$D^{-1}AV,$$

where $A = g(XW_Q W_K^\top X^\top / \sqrt{d})$ and $D := \mathrm{diag}(A\mathbf{1}_n)$.

As demonstrated in Theorem 1.1 of Kacham et al. (2024), one can construct a randomized feature map ϕ' for the degree-p polynomial kernel that ensures the resulting approximate attention weights remain non-negative, satisfy provable error bounds, and can be computed in time proportional to the sequence length n.

Theorem A.9 (Theorem 1.1 in Kacham et al. (2024)). *Let $p \geq 2$ be an even integer, $\epsilon \in (0, 0.5)$ be an error parameter. Let d be the dimension of the vectors to be mapped. There exists a randomized feature mapping*

$$\phi' : \mathbb{R}^d \to \mathbb{R}^{z^2}, \quad \text{for } z = \Theta\left(\frac{p}{\epsilon^2} \log \frac{1}{\delta}\right)$$

defined as $\phi'(x) := \left(Sx^{\otimes (p/2)}\right)^{\otimes 2}$ where $S \in \mathbb{R}^{z \times d^{p/2}}$ is a sketching matrix, such that for all set of vectors $\left\{q_i \in \mathbb{R}^d\right\}_{i \in [n]}$, $\left\{k_j \in \mathbb{R}^d\right\}_{j \in [n]}$, the following hold with probability at least $1 - \delta$:

1. *$\langle \phi'(q_i), \phi'(k_j) \rangle \geq 0$ for all $i, j \in [n]$;*
2. $$\sum_{i,j} |\langle \phi'(q_i), \phi'(k_j) \rangle - \langle q_i, k_j \rangle^p|^2 \leq \epsilon^2 \sum_{i,j} \|q_i\|_2^{2p} \|k_j\|_2^{2p};$$
3. *Computing $\phi'(x)$ for $x \in \mathbb{R}^d$ requires:*
   - *$\frac{p}{2}$ matrix-vector multiplications with matrices of size $d \times z$,*
   - *$\frac{p}{2} - 2$ matrix-vector multiplications with matrices of size $z \times z$,*
   - *$\frac{p}{2} - 1$ Hadamard products of z-dimensional vectors,*
   - *and 1 self-Kronecker product of an z-dimensional vector.*

To analyze the concentration and tail behavior of random variables—particularly in the context of sketching and randomized linear algebra—it is useful to work with their moment norms. The following definition introduces the L_t norm of a real-valued random variable, which captures its t-th moment in a normalized form. These norms play a key role in bounding approximation errors and analyzing stability under random projections. An important property of L_t norms is that they satisfy the triangle inequality, as guaranteed by the Minkowski inequality (Ahle et al., 2020).

Definition A.10 (Definition 4.1 in Ahle et al. (2020))**.** *For every integer $t \geq 1$ and any random variable $X \in \mathbb{R}$, we write*

$$\|X\|_{L_t} = \left(\mathbb{E}\left[|X|^t\right]\right)^{1/t}.$$

Note that $\|X+Y\|_{L_t} \leq \|X\|_{L_t} + \|Y\|_{L_t}$ for all random variables X, Y by the Minkowski Inequality.

Below, we present the JL moment property: it provides a probabilistic guarantee on how well a random matrix preserves the norm of any fixed unit vector—not just with high probability, but in expectation and in L_t norm. This property strengthens the classical JL lemma by quantifying how tightly concentrated the squared norm $\|Sx\|_2^2$ is around its mean.

Definition A.11 (JL Moment Property, Definition 4.2 in Ahle et al. (2020))**.** *For every positive integer t and every $\delta, \epsilon \geq 0$, we say a distribution over random matrices $S \in \mathbb{R}^{m \times d}$ has the (ϵ, δ, t)-JL-moment property, when*

$$\left\|\|Sx\|_2^2 - 1\right\|_{L_t} \leq \epsilon\delta^{1/t} \quad \text{and} \quad \mathbb{E}\left[\|Sx\|_2^2\right] = 1$$

for all $x \in \mathbb{R}^d$ such that $\|x\| = 1$.

By Ahle et al. (2020), the sketching matrix satisfying the JL Moment Property has the following property:

Lemma A.12 (Two vector JL Moment Property, Lemma 4.1 in Ahle et al. (2020))**.** *For all $x, y \in \mathbb{R}^d$, if S has the (ϵ, δ, t)-JL Moment Property, then*

$$\left\|(Sx)^\top (Sy) - x^\top y\right\|_{L_t} \leq \epsilon\delta^{1/t}\|x\|_2\|y\|_2.$$

B (BATCH) GAUSSIAN KERNEL DENSITY ESTIMATION VIA SINGLE THRESHOLD SUPPORT BASIS

In Appendix B.1, we present the definition of disjoint matrices and analyze their mathematical properties. In Appendix B.2, we give the formal definition of support basis and construct a single threshold support basis for $QK^\top$. In Appendix B.3, we generalize the bounded entry lemma (Alman and Song, 2023a) from the threshold $B = o\left(\sqrt{\log n}\right)$ to any arbitrary threshold being a positive real number. In Appendix B.4, we show that if the number of large entries in Q and K is small, then we can compute certain matrix multiplications in $A^{(L)}$ sparsity time. In Appendix B.5, we use our constructed single threshold support basis to design an $A^{(L)}$ sparsity time algorithm to approximate the (batch) Gaussian kernel density estimation AV.

B.1 Disjoint Matrices and Their Properties

In order to decompose the (Q, K)-softmax-attention matrix into simpler components, we begin by introducing the notion of disjoint matrices. This disjointness condition ensures that matrix components do not overlap in their non-zero entries, which is a crucial property that enables additive decompositions of matrix exponentials. We formalize this idea in the following definition.

Definition B.1 (Disjoint matrices)**.** *Let $\{A_k\}_{k\in[l]}$ be a set of n by n matrices, where $l \geq 2$ is an arbitrary positive integer. We say A_k's are disjoint matrices if for all $i, j \in [n]$, for all $k \in [l]$, for all $\overline{k} \in [l] \setminus \{k\}$, if $(A_k)_{i,j} \neq 0$, then $\left(A_{\overline{k}}\right)_{i,j} = 0$.*

Now, we show how the exponential of a sum of disjoint matrices can be simplified. Since disjoint matrices do not share any overlapping non-zero entries, the exponential of their sum equals the sum of their exponentials minus a constant matrix.

Fact B.2. *Let $B, C \in \mathbb{R}^{n\times n}$ be disjoint matrices (see Definition B.1).*

Then, we have

$$\exp(B+C) = \exp(B) + \exp(C) - \mathbf{1}_{n\times n}.$$

Proof. Without loss of generality, suppose $C_{i,j} \neq 0$ and $B_{i,j} = 0$. Then, for all $i, j \in [n]$, we have

$$\begin{aligned}\exp(B+C)_{i,j} &= \exp(B_{i,j} + C_{i,j}) \\ &= \exp(C)_{i,j} + 1 - 1 \\ &= \exp(C)_{i,j} + \exp(B)_{i,j} - (\mathbf{1}_{n\times n})_{i,j},\end{aligned}$$

where the first step follows from the fact that exp is applied entry-wisely, the second step follows from $B_{i,j} = 0$, and the last step follows from $\exp(0) = 1$. □

B.2 Support Basis

To construct a support basis for the matrix $QK^\top$, we must first define how to partition the matrix into "large" and "small" components based on the entries in Q and K.

Definition B.3. *Let $T > 0$ denote a threshold, $\alpha \in (0,1)$, and $C > 1$ denote a fixed constant. Let $Q, K \in \mathbb{R}^{n\times d}$ be the query and key matrices, respectively. We decompose these matrices as $Q = Q^{(L)} + Q^{(s)}$ and $K = K^{(L)} + K^{(s)}$, where:*

- $\left|\mathrm{supp}\left(Q^{(L)}\right)\right|, \left|\mathrm{supp}\left(K^{(L)}\right)\right| \leq Cn^\alpha$, *and for all $i \in [n]$ and $j \in [d]$, if $|Q_{i,j}| > T$, then $Q^{(L)}_{i,j} := Q_{i,j}$; otherwise, $Q^{(L)}_{i,j} := 0$. We define $K^{(L)}$ in a similar manner.*
- *For all $i \in [n]$ and $j \in [d]$, if $|Q_{i,j}| \leq T$, then $Q^{(s)}_{i,j} := Q_{i,j}$; otherwise, $Q^{(s)}_{i,j} := 0$. We define $K^{(s)}$ in a similar manner.*

We define $A^{(L)}$ and $A^{(s)}$ as follows:

Definition B.4. *Given the query and the key matrices $Q, K \in \mathbb{R}^{n\times d}$, we let $Q^{(L)}, Q^{(s)}, K^{(L)}, K^{(s)} \in \mathbb{R}^{n\times d}$ be defined as in Definition B.3. Let $i, j \in [n]$. We define matrices $A^{(L)}$ and $A^{(s)}$ as follows:*

$$A^{(L)}_{i,j} := \begin{cases} \left(QK^\top\right)_{i,j} & \text{if } \left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top\right)_{i,j} \neq 0 \\ 0 & \text{otherwise.}\end{cases}$$

$$A^{(s)}_{i,j} := \begin{cases} 0 & \text{if } \left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top\right)_{i,j} \neq 0 \\ \left(Q^{(s)}\left(K^{(s)}\right)^\top\right)_{i,j} & \text{otherwise.}\end{cases}$$

Now, we provide the formal definition of the support basis. For a given matrix A, it requires all matrices in this basis to be disjoint (see Definition B.1) to ensure the additive decomposition (see Fact B.2) and the sum of all matrices in this basis to be equal to the given matrix A.

Definition B.5 (Support basis)**.** *Let $\{A_k\}_{k\in[l]}$ be a set of n by n matrices, where $l \geq 2$ is an arbitrary positive integer. Given a matrix $A \in \mathbb{R}^{n\times n}$, we say that $\{A_k\}_{k\in[l]}$ is a support basis of A if*

1. *$\sum_{k\in[l]} A_k = A$ and*
2. *A_k's are disjoint matrices (see Definition B.1).*

Below, we formally prove that the resulting matrices $A^{(s)}$ and $A^{(L)}$ (as defined in Definition B.4) are disjoint and together sum to $QK^\top$, forming a valid support basis of $QK^\top$.

Lemma B.6. *Given the query and the key matrices $Q, K \in \mathbb{R}^{n \times d}$, we let $Q^{(L)}, Q^{(s)}, K^{(L)}, K^{(s)} \in \mathbb{R}^{n \times d}$ be defined as in Definition B.3 and $A^{(L)}, A^{(s)} \in \mathbb{R}^{n \times n}$ be defined as in Definition B.4.*

Then, $\left\{A^{(L)}, A^{(s)}\right\}$ is a support basis of the matrix $QK^\top \in \mathbb{R}^{n \times n}$.

Proof. We first show the first condition of the support basis, namely $A^{(s)} + A^{(L)} = QK^\top$.

By definition B.4, we can see that for all arbitrary $i, j \in [n]$, if

$$\left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top\right)_{i,j}$$

is not zero then:

$$A^{(s)}_{i,j} + A^{(L)}_{i,j} = \left(QK^\top\right)_{i,j} + 0 = \left(QK^\top\right)_{i,j}.$$

On the other hand, if $\left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top\right)_{i,j}$ is zero then we have:

$$\begin{aligned} A^{(s)}_{i,j} + A^{(L)}_{i,j} &= 0 + \left(Q^{(s)}\left(K^{(s)}\right)^\top\right)_{i,j} \\ &= \left(QK^\top\right)_{i,j} - \left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top\right)_{i,j} \\ &= \left(QK^\top\right)_{ij} + 0 \\ &= \left(QK^\top\right)_{ij}, \end{aligned}$$

where in the second equality, we use $QK^\top = \left(Q^{(L)} + Q^{(s)}\right)\left(K^{(L)} + K^{(s)}\right)^\top$.

Now, we show the second condition of the support basis that $A^{(s)}$ and $A^{(L)}$ are disjoint matrices.

This follows directly from Definition B.4: if $\left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top\right)_{i,j} \neq 0$, then $A^{(L)}_{i,j} = \left(QK^\top\right)_{i,j}$ and $A^{(s)}_{i,j} = 0$; otherwise $A^{(L)}_{i,j} = 0$ and $A^{(s)}_{i,j} = \left(Q^{(s)}\left(K^{(s)}\right)^\top\right)_{i,j}$. □

Now, we provide a concrete example of the support basis to better illustrate this decomposition.

Example B.7 (The support basis $\left\{A^{(L)}, A^{(s)}\right\}$ of $QK^\top$)**.** *Let $Q = \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \in \mathbb{R}^{3 \times 2}$ and $K^\top = \begin{bmatrix} g & i & k \\ h & j & l \end{bmatrix} \in \mathbb{R}^{2 \times 3}$. With a given threshold $T > 0$, we have $c, k > T$, and all other entries of Q and K are smaller than T. Therefore, by Definition B.3, we have*

$$\underbrace{\begin{bmatrix} 0 & 0 \\ c & 0 \\ 0 & 0 \end{bmatrix}}_{Q^{(L)}} + \underbrace{\begin{bmatrix} a & b \\ 0 & d \\ e & f \end{bmatrix}}_{Q^{(s)}} = \underbrace{\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}}_{Q} \quad \text{and} \quad \underbrace{\begin{bmatrix} 0 & 0 & k \\ 0 & 0 & 0 \end{bmatrix}}_{\left(K^{(L)}\right)^\top} + \underbrace{\begin{bmatrix} g & i & 0 \\ h & j & l \end{bmatrix}}_{\left(K^{(s)}\right)^\top} = \underbrace{\begin{bmatrix} g & i & k \\ h & j & l \end{bmatrix}}_{K^\top}$$

This setup allows us to decompose the matrix product $QK^\top$ into a sum of disjoint components:

$$\begin{aligned} & \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \begin{bmatrix} g & i & k \\ h & j & l \end{bmatrix} \\ = & \begin{bmatrix} ag + bh & ai + bj & ak + bl \\ cg + dh & ci + dj & ck + dl \\ eg + hf & ei + fj & ek + fl \end{bmatrix} \end{aligned}$$

$$
\begin{aligned}
&= \underbrace{\begin{bmatrix} 0 & 0 & ak+bl \\ cg+dh & ci+dj & ck+dl \\ 0 & 0 & ek+fl \end{bmatrix}}_{A^{(L)}} + \underbrace{\begin{bmatrix} ag+bh & ai+bj & 0 \\ 0 & 0 & 0 \\ eg+hf & ei+fj & 0 \end{bmatrix}}_{A^{(s)}} \\
&= \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & ck \\ 0 & 0 & 0 \end{bmatrix}}_{Q^{(L)}\left(K^{(L)}\right)^\top} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ cg & ci & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{Q^{(L)}\left(K^{(s)}\right)^\top} + \underbrace{\begin{bmatrix} 0 & 0 & ak \\ 0 & 0 & 0 \\ 0 & 0 & ek \end{bmatrix}}_{Q^{(s)}\left(K^{(L)}\right)^\top} + \underbrace{\begin{bmatrix} ag+bh & ai+bj & bl \\ dh & dj & dl \\ eg+hf & ei+fj & fl \end{bmatrix}}_{Q^{(s)}\left(K^{(s)}\right)^\top} \\
&= \underbrace{\begin{bmatrix} 0 & 0 \\ c & 0 \\ 0 & 0 \end{bmatrix}}_{Q^{(L)}} \underbrace{\begin{bmatrix} 0 & 0 & k \\ 0 & 0 & 0 \end{bmatrix}}_{\left(K^{(L)}\right)^\top} + \underbrace{\begin{bmatrix} 0 & 0 \\ c & 0 \\ 0 & 0 \end{bmatrix}}_{Q^{(L)}} \underbrace{\begin{bmatrix} g & i & 0 \\ h & j & l \end{bmatrix}}_{\left(K^{(s)}\right)^\top} + \underbrace{\begin{bmatrix} a & b \\ 0 & d \\ e & f \end{bmatrix}}_{Q^{(s)}} \underbrace{\begin{bmatrix} 0 & 0 & k \\ 0 & 0 & 0 \end{bmatrix}}_{\left(K^{(L)}\right)^\top} + \underbrace{\begin{bmatrix} a & b \\ 0 & d \\ e & f \end{bmatrix}}_{Q^{(s)}} \underbrace{\begin{bmatrix} g & i & 0 \\ h & j & l \end{bmatrix}}_{\left(K^{(s)}\right)^\top}.
\end{aligned}
$$

This example illustrates how the product $QK^\top$ can be partitioned into disjoint matrix components, forming a support basis that enables structured approximation.

B.3 Bounded Entry

We now establish an upper bound on the entries of the matrix $A^{(s)}$, which corresponds to the contribution from the small entries of Q and K. This allows us to apply polynomial approximation techniques with controlled error. We generalize the bounded-entry result from Alman and Song (2023a) to arbitrary thresholds.

Lemma B.8 (Bounded entry, an improved version of Lemma 3.3 in Alman and Song (2023a))**.** *Given $Q, K \in \mathbb{R}^{n \times d}$, we let $Q^{(s)}, K^{(s)}$ be defined as in Definition B.3. Let $A^{(s)} \in \mathbb{R}^{n \times n}$ be defined as in Definition B.4, where*

$$
A^{(s)}_{i,j} := \begin{cases} 0 & \text{if } \left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top\right)_{i,j} \neq 0 \\ \left(Q^{(s)}\left(K^{(s)}\right)^\top\right)_{i,j} & \text{otherwise.} \end{cases}
$$

Then, for a given threshold $T > 0$, we have

$$
\left\| A^{(s)}/d \right\|_\infty \leq T^2.
$$

Proof. We have

$$
\begin{aligned}
\left\| A^{(s)} \right\|_\infty &= \max_{(i,j)\in[n]\times[n]} \left| A^{(s)}_{i,j} \right| \\
&= \max_{(i,j)\in[n]\times[n]} \left| \sum_{l=1}^d Q^{(s)}_{i,l} K^{(s)}_{j,l} \right| \\
&\leq \sum_{l=1}^d \left| \max_{i\in[n]} Q^{(s)}_{i,l} \right| \cdot \left| \max_{j\in[n]} K^{(s)}_{j,l} \right| \\
&\leq \sum_{l=1}^d T^2 \\
&= dT^2,
\end{aligned}
$$

where the first step follows from the definition of the ℓ_∞ norm, the second step follows from the definition of $A^{(s)}$ (see Definition B.4), the third step follows from the triangle inequality (see Fact J.3), and the fourth step follows from Definition B.3. □

B.4 Running Time and Sparsity Analysis

Next, we analyze the sparsity and computational cost associated with the matrix $A^{(L)}$, which captures the contribution from the large entries of Q and K. Our goal is to exactly compute the part of the attention

computation involving $A^{(L)}$ and approximate the part involving $A^{(s)}$. Therefore, it is important to know the time complexity for constructing $A^{(L)}$ and its sparsity.

Lemma B.9. *Let $\alpha \in (0,1)$. Given the query and the key matrices $Q, K \in \mathbb{R}^{n \times d}$, we let $Q^{(L)}, Q^{(s)}, K^{(L)}, K^{(s)} \in \mathbb{R}^{n \times d}$ be defined as in Definition B.3 and $A^{(L)}, A^{(s)} \in \mathbb{R}^{n \times d}$ be defined as in Definition B.4.*

Then, we have

1. $\left|\mathrm{supp}\left(A^{(L)}\right)\right| = O\left(n^{1+\alpha}\right)$, *and*
2. *it takes $O\left(n^{1+\alpha} d\right)$ time to compute $A^{(L)}$.*

Proof. **Proof of Part 1.**

Note that by Definition B.4, we have

$$A_{i,j}^{(L)} := \begin{cases} \left(QK^\top\right)_{i,j} & \text{if } \left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top\right)_{i,j} \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Suppose $(i,j) \notin \mathrm{supp}\left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top\right)$.

Then, we have

$$\begin{aligned} A_{i,j}^{(L)} = & \left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top\right)_{i,j} \\ = & 0. \end{aligned}$$

Therefore, it suffices to consider the case when

$$(i,j) \in \mathrm{supp}\left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top\right).$$

Note that by Definition B.3, we have

$$\left|\mathrm{supp}\left(Q^{(L)}\right)\right|, \left|\mathrm{supp}\left(K^{(L)}\right)\right| \leq C n^{\alpha}, \tag{6}$$

for a fixed constant $C > 1$.

Therefore, we have

$$\begin{aligned} & \left|\mathrm{supp}\left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top\right)\right| \\ \leq & \left|\mathrm{supp}\left(Q^{(L)}\left(K^{(L)}\right)^\top\right)\right| + \left|\mathrm{supp}\left(Q^{(s)}\left(K^{(L)}\right)^\top\right)\right| + \left|\mathrm{supp}\left(Q^{(L)}\left(K^{(s)}\right)^\top\right)\right| \\ \leq & O\left(n^{2\alpha}\right) + \left|\mathrm{supp}\left(Q^{(s)}\left(K^{(L)}\right)^\top\right)\right| + \left|\mathrm{supp}\left(Q^{(L)}\left(K^{(s)}\right)^\top\right)\right| \\ \leq & O\left(n^{2\alpha}\right) + O\left(n^{1+\alpha}\right) + O\left(n^{1+\alpha}\right) \\ = & O\left(n^{1+\alpha}\right), \end{aligned} \tag{7}$$

where the first step follows from Fact J.1, the second step follows from Eq. (6), the third step follows from Eq. (6), and the last step follows from $\alpha \in (0,1)$.

Proof of Part 2.

The proof of this part is similar to that of **Part 1**.

It also suffices to only consider the case when

$$(i,j) \in \mathrm{supp}\left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top\right).$$

Because of Eq. (6), we have that

- computing $Q^{(L)}\left(K^{(L)}\right)^\top$ takes $O\left(n^{2\alpha}\right)$ time,
- computing $Q^{(L)}\left(K^{(s)}\right)^\top$ takes $O\left(n^{1+\alpha}\right)$ time, and
- computing $Q^{(s)}\left(K^{(L)}\right)^\top$ takes $O\left(n^{1+\alpha}\right)$ time.

Moreover, because of Eq. (7), we need to compute $O\left(n^{1+\alpha}\right)$ numbers of entries of $Q^{(s)}\left(K^{(s)}\right)^\top$.

Since $Q^{(s)}, K^{(s)} \in \mathbb{R}^{n\times d}$, for each $i,j \in [n]$, we have

$$\left(Q^{(s)}\left(K^{(s)}\right)^\top\right)_{i,j} = \sum_{k\in[d]}\left(Q^{(s)}\right)_{i,k}\left(K^{(s)}\right)^\top_{k,j},$$

which takes $O(d)$ time.

Therefore, in total, it takes $O\left(n^{1+\alpha}d\right)$ times to compute $\left(Q^{(s)}\left(K^{(s)}\right)^\top\right)_{i,j}$ for all

$$(i,j) \in \mathrm{supp}\left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top\right).$$

Therefore, the time complexity of computing

$$\left(QK^\top\right)_{i,j} = \left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top + Q^{(s)}\left(K^{(s)}\right)^\top\right)_{i,j}$$

for all $(i,j) \in \mathrm{supp}\left(Q^{(L)}\left(K^{(L)}\right)^\top + Q^{(s)}\left(K^{(L)}\right)^\top + Q^{(L)}\left(K^{(s)}\right)^\top\right)$ is

$$O\left(n^{2\alpha}\right) + O\left(n^{1+\alpha}\right) + O\left(n^{1+\alpha}\right) + O\left(n^{1+\alpha}d\right) = O\left(n^{1+\alpha}d\right).$$

□

B.5 (Batch) Gaussian Kernel Density Estimation

We now formalize the task of computing AV, where $A = \exp(QK^\top/d)$ is the (Q,K)-softmax-attention matrix. This operation is closely related to a classical technique in statistics known as *Gaussian Kernel Density Estimation (KDE)*. In Gaussian KDE, the goal is to estimate a density function by averaging Gaussian kernels centered at given data points. Given data points $\{x_1,\ldots,x_n\} \subset \mathbb{R}^d$ and a query point $q \in \mathbb{R}^d$, the classical Gaussian KDE estimate is:

$$\widehat{f}(q) = \frac{1}{n}\sum_{j=1}^{n}\exp\left(-\frac{\|q-x_j\|_2^2}{2h^2}\right),$$

where the Gaussian kernel can be expressed as:

$$\exp\left(-\frac{\|q-k\|_2^2}{2h^2}\right) = \exp\left(-\frac{\|q\|_2^2+\|k\|_2^2-2\langle q,k\rangle}{2h^2}\right),$$

and the attention computation problem focus primarily on

$$\exp\left(\frac{\langle q,k\rangle}{h^2}\right).$$

The matrix product AV can therefore be interpreted as a *batch* version of Gaussian KDE, where each row of AV aggregates value vectors $V_{*,k}$ weighted by the similarity between query $Q_{i,*}$ and key $K_{j,*}$. This motivates the name *(Batch) Gaussian Kernel Density Estimation*, as it generalizes Gaussian KDE to the matrix setting. We now formally define the problem.

Definition B.10 ((Batch) Gaussian kernel density estimation)**.** *Given the query $Q \in \mathbb{R}^{n \times d}$, key $K \in \mathbb{R}^{n \times d}$, and value $V \in \mathbb{R}^{n \times d}$, we define the (Q, K)-softmax-attention matrix as $A = \exp\left(QK^\top/d\right)$. The goal of the (batch) Gaussian kernel density estimation is to output $\mathsf{S} \in \mathbb{R}^{n \times d}$ satisfying, for all $\epsilon > 0$,*

$$\|\mathsf{S} - AV\|_\infty < \epsilon.$$

Having defined the (Batch) Gaussian Kernel Density Estimation problem, we now show how our algorithm can approximate AV efficiently by leveraging the support basis decomposition. Specifically, we use polynomial approximation for the component $A^{(s)}$, which contains only small entries and is thus suitable for Chebyshev expansion, and we compute the remaining part $A^{(L)}$ explicitly, exploiting its sparsity. The following lemma guarantees the correctness and efficiency of this approach by bounding the approximation error and analyzing the overall runtime.

Algorithm 3 In this algorithm, we approximate the (batch) Gaussian kernel density estimation: given the query, key, and value matrices $Q \in \mathbb{R}^{n \times d}, K \in \mathbb{R}^{n \times d}, V \in \mathbb{R}^{n \times d}$, the goal is to output $\widetilde{A}V$, where $\widetilde{A} \in \mathbb{R}^{n \times n}$ is the (ϵ, r)-approximation of the (Q, K)-softmax-attention matrix A.

procedure GaussianKDE($Q \in \mathbb{R}^{n \times d}, K \in \mathbb{R}^{n \times d}, V \in \mathbb{R}^{n \times d}, n, d, \epsilon, T = o\left(\sqrt{\log n}\right)$)
  $Q^{(L)}, Q^{(s)}, K^{(L)}, K^{(s)} \in \mathbb{R}^{n \times d} \leftarrow$ Split(Q, T), Split(K, T)
  Explicitly compute $A^{(L)}$.
  $U_1, U_2 \in \mathbb{R}^{n \times r} \leftarrow$ Polynomial$(Q^{(s)}, K^{(s)}, \epsilon_0)$. ▷ To approximate $\exp\left(A^{(s)}/d\right)$.
  Get $C_1 \leftarrow U_1 \left(U_2^\top V\right) \in \mathbb{R}^{n \times d}$.
  Get $C_2 \leftarrow \left(\exp\left(A^{(L)}/d\right) - \mathbf{1}_{n \times n}\right) V \in \mathbb{R}^{n \times d}$.
  return $C_1 + C_2 \in \mathbb{R}^{n \times d}$.
end procedure

Lemma B.11. *Given the query $Q \in \mathbb{R}^{n \times d}$, key $K \in \mathbb{R}^{n \times d}$, and value $V \in \mathbb{R}^{n \times d}$, we define the (Q, K)-softmax-attention matrix as $A = \exp\left(QK^\top/d\right)$ and $Q^{(L)}, Q^{(s)}, K^{(L)}, K^{(s)} \in \mathbb{R}^{n \times d}$ as in Definition B.3. Suppose by Lemma A.4, there exists $U_1, U_2 \in \mathbb{R}^{n \times r}$ such that $U_1 U_2^\top$ is the (ϵ_0, r)-approximation of $\exp\left(Q^{(s)}\left(K^{(s)}\right)^\top/d\right)$ for all arbitrary $\epsilon_0 \in (0, 0.1)$.*

Then, we can solve the (Batch) Gaussian kernel density estimation (Definition B.10) by outputting $\mathsf{S} \in \mathbb{R}^{n \times d}$ (Algorithm 3) satisfying

$$\|\mathsf{S} - AV\|_\infty < n\epsilon_0 \|V\|_\infty$$

in $O\left(n^{1+\alpha} d\right)$ time.

Proof. **Proof of correctness.**

We have

$$\begin{aligned} AV =& \exp\left(QK^\top/d\right) V \\ =& \exp\left(\left(A^{(s)} + A^{(L)}\right)/d\right) V \\ =& \exp\left(A^{(s)}/d\right) V + \exp\left(A^{(L)}/d\right) V - \mathbf{1}_{n \times n} V, \end{aligned} \tag{8}$$

where the first step follows from the definition of the (Q, K)-softmax-attention matrix, the second step follows from Lemma B.6, and the third step follows from Fact B.2.

By the Lemma B.8, we have

$$\left\|A^{(s)}/d\right\|_\infty \leq T^2,$$

for some threshold $T > 0$.

By setting the threshold T to be equal to $B = o\left(\sqrt{\log n}\right)$, we can see that this satisfies the assumption of using Lemma A.4 and Algorithm 2. Therefore, using Lemma A.4, there exists $U_1, U_2 \in \mathbb{R}^{n \times r}$ such that $U_1 U_2^\top$ is the (ϵ_0, r)-approximation of $\exp\left(Q^{(s)} \left(K^{(s)}\right)^\top / d\right)$, satisfying

$$\left\| \exp\left(A^{(s)}/d\right) - U_1 U_2^\top \right\|_\infty < \epsilon_0, \tag{9}$$

for all arbitrary $\epsilon_0 \in (0, 0.1)$.

In Algorithm 3, we output

$$\begin{aligned} \mathsf{S} &= C_1 + C_2 \\ &= U_1 \left(U_2^\top V\right) + \left(\exp\left(A^{(L)}/d\right) - \mathbf{1}_{n \times n}\right) V \\ &= U_1 \left(U_2^\top V\right) + \exp\left(A^{(L)}/d\right) V - \mathbf{1}_{n \times n} V, \end{aligned} \tag{10}$$

where the first step follows from the output (Line 7) of Algorithm 3 and the second step follows from updates $C_1 \leftarrow U_1 \left(U_2^\top V\right) \in \mathbb{R}^{n \times d}$ (Line 5) and $C_2 \leftarrow \left(\exp\left(A^{(L)}/d\right) - \mathbf{1}_{n \times n}\right) V \in \mathbb{R}^{n \times d}$ (Line 6).

Therefore, combining Eq. (8) and Eq. (10), we have

$$\begin{aligned} \|\mathsf{S} - AV\|_\infty = & \left\| U_1 U_2^\top V - \exp\left(A^{(s)}/d\right) V \right\|_\infty \\ = & \left\| \left(U_1 U_2^\top - \exp\left(A^{(s)}/d\right)\right) V \right\|_\infty \\ \leq & n \left\| U_1 U_2^\top - \exp\left(A^{(s)}/d\right) \right\|_\infty \|V\|_\infty \\ \leq & n\epsilon_0 \|V\|_\infty, \end{aligned}$$

where the first step follows from combining Eq. (10) and Eq. (8), the second step follows from the distributive law, the third step follows from the definition of the ℓ_∞ norm, and the fourth step follows from Eq. (9).

Proof of the running time.

In Line 2, we need to check each entry of $Q, K \in \mathbb{R}^{n \times d}$ finding the ones greater than the threshold T, which takes $O(nd)$ time.

In Line 3, we compute $A^{(L)} \in \mathbb{R}^{n \times n}$. By **Part 2** of Lemma B.9, we know that it takes $O\left(n^{1+\alpha} d\right)$ time to compute $A^{(L)}$.

In Line 4, we construct $U_1, U_2 \in \mathbb{R}^{n \times r}$, which by Lemma A.4 takes $O(nr)$ time.

In Line 5, it takes $O(nrd)$ time to compute $U_2^\top V \in \mathbb{R}^{r \times d}$ and takes $O(nrd)$ time to compute $U_1 \left(U_2^\top V\right) \in \mathbb{R}^{n \times d}$.

In Line 6, it takes $O(n^{1+\alpha} d)$ time to compute $\left(\exp\left(A^{(L)}/d\right) - \mathbf{1}_{n \times n}\right) V$ as by **Part 1** of Lemma B.9, we know that $\left|\mathrm{supp}\left(A^{(L)}\right)\right| = O\left(n^{1+\alpha}\right)$ which implies $\left|\mathrm{supp}\left(\exp\left(A^{(L)}/d\right) - \mathbf{1}_{n \times n}\right)\right| = O\left(n^{1+\alpha}\right)$.

In total, it takes $O(n^{1+\alpha} d + nrd)$ time. By Lemma A.4, we know that $O(nr) = n^{1+o(1)}$. Therefore, we have

$$O(n^{1+\alpha} d + nrd) = O(n^{1+\alpha} d)$$

□

C ATTENTION OPTIMIZATION VIA SINGLE THRESHOLD SUPPORT BASIS

In Appendix C.1, we cite a lemma from Alman and Song (2023a) stating that if we can have the relative error between the (Q, K)-softmax-attention matrix A and the approximate (Q, K)-softmax-attention matrix $\widetilde{A}$, then we can find the relative error between $\mathrm{diag}(A\mathbf{1}_n)$ and $\mathrm{diag}(\widetilde{A}\mathbf{1}_n)$. In Appendix C.2, we get the relative error

between $\mathrm{diag}(A\mathbf{1}_n)$ and $\mathrm{diag}(\widetilde{A}\mathbf{1}_n)$ by finding the relative error between A and $\widetilde{A}$. In Appendix C.3, we combine these relative errors with our result on the (batch) Gaussian kernel density estimation (Lemma B.11) to construct an $A^{(L)}$ sparsity time algorithm to approximate the attention computation.

C.1 Relationship Between the Attention Matrix and the Normalization Matrix

We cite Lemma 3.5 from Alman and Song (2023a) which states that if we can get $\left|\widetilde{A}_{i,j} - A_{i,j}\right| \leq \epsilon_A \cdot A_{i,j}$, then we have $\left|\widetilde{D}_{i,i} - D_{i,i}\right| \leq \epsilon_A \cdot D_{i,i}$.

Lemma C.1 (Lemma 3.5 in Alman and Song (2023a))**.** *Let $A \in \mathbb{R}^{n \times n}$ be any matrix whose entries are all positive and $\epsilon_A \in (0, 0.1)$ be any parameter. Let $\widetilde{A} \in \mathbb{R}^{n \times n}$ be any matrix such that, for all $(i,j) \in [n] \times [n]$, we have*

$$\left|\widetilde{A}_{i,j} - A_{i,j}\right| \leq \epsilon_A \cdot A_{i,j}.$$

Define the matrices $D, \widetilde{D} \in \mathbb{R}^{n \times n}$ by $D = \mathrm{diag}(A\mathbf{1}_n)$ and $\widetilde{D} = \mathrm{diag}(\widetilde{A}\mathbf{1}_n)$. Then, for all $i \in [n]$ we have

$$\left|\widetilde{D}_{i,i} - D_{i,i}\right| \leq \epsilon_A \cdot D_{i,i}.$$

Proof. We have

$$\begin{aligned}
\left|\widetilde{D}_{i,i} - D_{i,i}\right| &= \left|\sum_{j=1}^n \widetilde{A}_{i,j} - \sum_{j=1}^n A_{i,j}\right| \\
&\leq \sum_{j=1}^n \left|\widetilde{A}_{i,j} - A_{i,j}\right| \\
&\leq \sum_{j=1}^n \epsilon_A A_{i,j} \\
&= \epsilon_A \cdot D_{i,i},
\end{aligned}$$

where the first step follows from $D = \mathrm{diag}(A\mathbf{1}_n)$ and $\widetilde{D} = \mathrm{diag}(\widetilde{A}\mathbf{1}_n)$, the second step follows from the triangle inequality (Fact J.3), the third step follows from the assumption in the lemma statement $\left|\widetilde{A}_{i,j} - A_{i,j}\right| \leq \epsilon_A \cdot A_{i,j}$, and the last step follows from $D = \mathrm{diag}(A\mathbf{1}_n)$. □

C.2 The Error Bound for the Normalization Matrix

Now, we prove $\left|\widetilde{A}_{i,j} - A_{i,j}\right| \leq \epsilon_A \cdot A_{i,j}$ so that by the logic of Lemma C.1, we can get $\left|\widetilde{D}_{i,i} - D_{i,i}\right| \leq \epsilon \cdot D_{i,i}$.

Lemma C.2. *Given the query $Q \in \mathbb{R}^{n \times d}$ and the key $K \in \mathbb{R}^{n \times d}$, we define the (Q,K)-softmax-attention matrix as $A = \exp\left(QK^\top/d\right)$ and $Q^{(L)}, Q^{(s)}, K^{(L)}, K^{(s)} \in \mathbb{R}^{n \times d}$ as in Definition B.3. Suppose by Lemma A.4, there exists $U_1, U_2 \in \mathbb{R}^{n \times r}$ such that $U_1 U_2^\top$ is the (ϵ_0, r)-approximation of $\exp\left(Q^{(s)}\left(K^{(s)}\right)^\top/d\right)$ for all arbitrary $\epsilon_0 \in (0, 0.1)$. With $\widetilde{A} = U_1 U_2^\top + \exp\left(A^{(L)}/d\right) - \mathbf{1}_{n \times n}$, we define $\widetilde{D} := \mathrm{diag}\left(\widetilde{A}\mathbf{1}_n\right)$ and $D := \mathrm{diag}\left(A\mathbf{1}_n\right)$.*

Then, we can show that for all $\epsilon \in (0, 0.1)$ and $i \in [n]$,

$$\left|\widetilde{D}_{i,i} - D_{i,i}\right| \leq \epsilon \cdot D_{i,i}.$$

Proof. First, for all $i, j \in [n]$, we have

$$\begin{aligned}
\left|\widetilde{A}_{i,j} - A_{i,j}\right| &= \left|\left(\exp\left(A^{(s)}/d\right) + \exp\left(A^{(L)}/d\right) - \mathbf{1}_{n \times n}\right)_{i,j} - \left(U_1 U_2^\top + \exp\left(A^{(L)}/d\right) - \mathbf{1}_{n \times n}\right)_{i,j}\right| \\
&= \left|\left(\exp\left(A^{(s)}/d\right) + \exp\left(A^{(L)}/d\right) - \mathbf{1}_{n \times n} - U_1 U_2^\top - \exp\left(A^{(L)}/d\right) + \mathbf{1}_{n \times n}\right)_{i,j}\right|
\end{aligned}$$

$$= \left| \left(\exp\left(A^{(s)}/d \right) - U_1 U_2^\top \right)_{i,j} \right|$$
$$\leq \epsilon \left(\exp\left(A^{(s)}/d \right) \right)_{i,j},$$

where the first step follows from $A = \exp\left(QK^\top/d\right) = \exp\left(A^{(s)}/d\right) + \exp\left(A^{(L)}/d\right) - \mathbf{1}_{n\times n}$ (see Eq. (8) for detail) and $\widetilde{A} = U_1 U_2^\top + \exp\left(A^{(L)}/d\right) - \mathbf{1}_{n\times n}$ (see from the lemma statement) and the last step follows from Lemma A.4.

Therefore, by Lemma C.1, we have for all $i \in [n]$,

$$\left| \widetilde{D}_{i,i} - D_{i,i} \right| \leq \epsilon \cdot D_{i,i}.$$

□

C.3 Main Result

We now state our main theoretical result for the single-threshold support basis framework. Building on Lemma B.11 and Lemma C.2, we show that the entire attention computation $D^{-1}AV$ can be approximated in $A^{(L)}$ sparsity time with a relative error guarantee. Our approach combines two key components: a polynomial approximation for the bounded portion of the (Q, K)-softmax-attention matrix, and an explicit computation for the sparse, large entries. Together, these enable us to construct an approximate attention output P such that

$$\|P - D^{-1}AV\|_\infty < \epsilon \cdot \|V\|_\infty,$$

for all accuracy $\epsilon > 0$. The following theorem formalizes this guarantee and provides the corresponding runtime bound.

Theorem C.3. *Given the query $Q \in \mathbb{R}^{n\times d}$, key $K \in \mathbb{R}^{n\times d}$, and value $V \in \mathbb{R}^{n\times d}$, we define the (Q, K)-softmax-attention matrix as $A = \exp\left(QK^\top/d\right)$ and $Q^{(L)}, Q^{(s)}, K^{(L)}, K^{(s)} \in \mathbb{R}^{n\times d}$ as in Definition B.3. Let $\epsilon \in (0, 0.1)$.*

Then, we can solve the approximate attention computation (Definition 1.2) by outputting $P \in \mathbb{R}^{n\times d}$ (Algorithm 1) satisfying

$$\left\|P - D^{-1}AV\right\|_\infty < \epsilon \|V\|_\infty$$

in $O(n^{1+\alpha}d)$ time.

Proof. **Proof of correctness.**

By the triangle inequality (see Fact J.3), we have

$$\begin{aligned} \left\| D^{-1}AV - \widetilde{D}^{-1}\widetilde{A}V \right\|_\infty &= \left\| D^{-1}AV - D^{-1}\widetilde{A}V + D^{-1}\widetilde{A}V - \widetilde{D}^{-1}\widetilde{A}V \right\|_\infty \\ &\leq \left\| D^{-1}AV - D^{-1}\widetilde{A}V \right\|_\infty + \left\| D^{-1}\widetilde{A}V - \widetilde{D}^{-1}\widetilde{A}V \right\|_\infty. \end{aligned} \tag{11}$$

Similar as Alman and Song (2023a), we have for each $(i, j) \in [n] \times [d]$,

$$\begin{aligned} \left| (\widetilde{D}^{-1}\widetilde{A}V - D^{-1}\widetilde{A}V)_{i,j} \right| &= \left| \sum_{l=1}^n \left(\widetilde{D}_{i,i}^{-1} - D_{i,i}^{-1} \right) \cdot \widetilde{A}_{i,l} \cdot V_{l,j} \right| \\ &\leq \sum_{l=1}^n \left| \widetilde{D}_{i,i}^{-1} - D_{i,i}^{-1} \right| \cdot |\widetilde{A}_{i,l}| \cdot \|V\|_\infty \\ &= \sum_{l=1}^n \left| \frac{D_{i,i} - \widetilde{D}_{i,i}}{D_{i,i}\widetilde{D}_{i,i}} \right| \cdot |\widetilde{A}_{i,l}| \cdot \|V\|_\infty \\ &\leq \epsilon \cdot \sum_{l=1}^n \left| \widetilde{D}_{i,i}^{-1}\widetilde{A}_{i,l} \right| \cdot \|V\|_\infty \end{aligned}$$

$$= \epsilon \cdot \left| \sum_{l=1}^n \widetilde{D}_{i,i}^{-1} \widetilde{A}_{i,l} \right| \cdot \|V\|_\infty$$
$$= \epsilon \cdot \|V\|_\infty,$$

where the second step follows from the triangle inequality (see Fact J.3), the fourth step follows from Lemma C.2 that

$$\left| \frac{D_{i,i} - \widetilde{D}_{i,i}}{D_{i,i}} \right| \leq \epsilon,$$

the fifth step follows from $\widetilde{D}_{i,i}^{-1} > 0$ and $\widetilde{A}_{i,l} > 0$, and the last step follows from $\left| \sum_{l=1}^n \widetilde{D}_{i,i}^{-1} \widetilde{A}_{i,l} \right| = 1$ as $D = \mathrm{diag}(A\mathbf{1}_n)$.

Also, for each $(i,j) \in [n] \times [d]$,

$$\begin{aligned}
\left| (D^{-1}\widetilde{A}V - D^{-1}AV)_{i,j} \right| &= \left| \sum_{l=1}^n D_{i,i}^{-1} (\widetilde{A}_{i,l} - A_{i,l}) \cdot V_{l,j} \right| \\
&\leq \sum_{l=1}^n \left| D_{i,i}^{-1} \right| \cdot |\widetilde{A}_{i,l} - A_{i,l}| \cdot \|V\|_\infty \\
&= \sum_{l=1}^n D_{i,i}^{-1} \cdot |\widetilde{A}_{i,l} - A_{i,l}| \cdot \|V\|_\infty \\
&\leq \sum_{l=1}^n D_{i,i}^{-1} \cdot \epsilon A_{i,l} \cdot \|V\|_\infty \\
&= \epsilon \cdot \|V\|_\infty,
\end{aligned}$$

where the second step follows from triangle inequality (see Fact J.3), the third step follows from $D_{i,i}^{-1} > 0$, the fourth step follows from $|\widetilde{A}_{i,l} - A_{i,l}| \leq \epsilon \cdot A_{i,l}$ (see Lemma C.2), and the last step follows from definition of $D_{i,i}$.

Proof of Running time.

Similar as the proof of Lemma B.11, Line 4 to Line 8 takes $O(n^{1+\alpha} d)$ time, Line 9 to Line 10 take $O(n^{1+\alpha} d)$ time, and computing $D^{-1}(C_1 + C_2)$ takes $O(nd)$ time (as D^{-1} is a diagonal matrix). □

D ATTENTION OPTIMIZATION UNDER THE SUB-GAUSSIAN DISTRIBUTION

Now, we justify the sparsity assumption used in our single-threshold support basis decomposition by introducing a natural probabilistic model: the sub-Gaussian distribution. Empirically, the entries of the query and key matrices in large language models tend to concentrate near zero and exhibit light tails—properties well captured by sub-Gaussian random variables. Under this assumption, we show that the number of "large" entries in Q and K (those exceeding a chosen threshold) is small with high probability. This validates our decomposition strategy and ensures that the matrix $A^{(L)}$ remains sparse. We also analyze the expected number of large entries and their contribution to the (Q,K)-softmax-attention matrix, supporting the theoretical guarantees presented in Appendices B and C.

Specifically, in Appendix D.1, we present the formal definition of a sub-Gaussian distribution. In Appendix D.2, we assume that the entries of the query matrix Q and the key matrix K follow sub-Gaussian distributions and show that the expected number of entries exceeding any threshold $T > 0$ is bounded. In Appendix D.3, we prove that the number of non-zero entries in $Q^{(L)}$ and $K^{(L)}$ is at most n^α for some $\alpha \in (0,1)$, with high probability. Finally, in Appendix D.4, we combine this sparsity result with our main result on attention optimization (Theorem C.3) to show that, with high probability, the attention computation can be approximated in sub-quadratic time. Thus, we finally can formally justify that the $A^{(L)}$ sparsity time $O\left(n^{1+\alpha}\right)$ in Theorem C.3 is sub-quadratic in n.

D.1 The Definition of Sub-Gaussian Distribution

To formally analyze the statistical properties of the query and key matrices, we begin by introducing the notion of a sub-Gaussian distribution. Sub-Gaussian random variables exhibit tail behavior similar to or lighter than that of a Gaussian, making them a natural model for the entries of Q and K in practice. This assumption enables us to rigorously bound the probability and expected number of large entries, which is essential for proving sparsity and deriving sub-quadratic algorithms.

Definition D.1 (Proposition 2.5.2 in Vershynin (2018))**.** *A random variable X is said to be sub-Gaussian with variance proxy σ^2, denoted as $X \in \mathrm{subG}(\sigma^2)$, if it satisfies*

$$\Pr\left[|X| \geq t\right] \leq 2\exp\left(-\frac{t^2}{\sigma^2}\right) \quad \text{for all } t \geq 0.$$

D.2 The Expected Number of Large Entries

Under the sub-Gaussian assumption, we can now quantify how many entries in the query and key matrices exceed a given threshold. Specifically, the following lemma shows that the expected number of such large entries decays exponentially with the threshold. This result supports our sparsity assumption for $Q^{(L)}$ and $K^{(L)}$, and plays a key role in bounding the size of the matrix $A^{(L)}$ used in our support basis decomposition.

Lemma D.2. *Let $Q, K, V \in \mathbb{R}^{n\times d}$ be random matrices whose entries are independent and sub-Gaussian with variance proxies σ_Q^2, σ_K^2, and σ_V^2, respectively, which implies that for all $i \in [n]$, $j \in [d]$, and $t \geq 0$,*

$$\Pr\left[|Q_{i,j}| \geq t\right] \leq 2\exp\left(-\frac{t^2}{\sigma_Q^2}\right),$$

and similarly for $K_{i,j}$ and $V_{i,j}$. Define $Q^{(L)} \in \mathbb{R}^{n\times d}$ as

$$Q_{i,j}^{(L)} := \begin{cases} Q_{i,j} & \text{if } |Q_{i,j}| > T, \\ 0 & \text{otherwise,} \end{cases}$$

and similarly for $K^{(L)}, V^{(L)} \in \mathbb{R}^{n\times d}$. For all $M^{(L)} \in \{Q^{(L)}, K^{(L)}, V^{(L)}\}$ and $\sigma_M \in \{\sigma_Q, \sigma_K, \sigma_V\}$, if $d = O(\log n)$, $T = o(\sqrt{\log n})$, and $T = \omega(\sqrt{\log\log n})$, then

$$\mathbb{E}\left[\left|\mathrm{supp}(M^{(L)})\right|\right] \leq n^{1-o(1)}.$$

Proof. By linearity of expectation, we have

$$\begin{aligned}
\mathbb{E}\left[\left|\mathrm{supp}(M^{(L)})\right|\right] &= \sum_{i=1}^{n}\sum_{j=1}^{d}\Pr\left[|M_{i,j}| > T\right] \\
&\leq \sum_{i=1}^{n}\sum_{j=1}^{d} 2\exp\left(-\frac{T^2}{\sigma_M^2}\right) \\
&= 2nd\exp\left(-\frac{T^2}{\sigma_M^2}\right).
\end{aligned}$$

Since $d = O(\log n)$, there exist constants $C > 0$ and $n_0 \in \mathbb{Z}_+$ such that for all $n \geq n_0$, $d \leq C\log n$. Hence, for all sufficiently large n, we have

$$\mathbb{E}\left[\left|\mathrm{supp}(M^{(L)})\right|\right] \leq 2Cn\log n \cdot \exp\left(-\frac{T^2}{\sigma_M^2}\right).$$

We now rewrite the right-hand side in the form $n^{1-\varepsilon_n}$. Observe that $\log n = n^{\frac{\log\log n}{\log n}}$ and $\exp\left(-\frac{T^2}{\sigma_M^2}\right) = n^{-\frac{T^2}{\sigma_M^2\log n}}$. Therefore, we have

$$\mathbb{E}\left[\left|\mathrm{supp}(M^{(L)})\right|\right] \leq 2Cn^{1+\frac{\log\log n}{\log n}-\frac{T^2}{\sigma_M^2\log n}}$$

$$= n^{1+\frac{\log\log n}{\log n}-\frac{T^2}{\sigma_M^2\log n}+\frac{\log(2C)}{\log n}}$$

We define

$$\varepsilon_n := \frac{T^2}{\sigma_M^2\log n} - \frac{\log\log n}{\log n} - \frac{\log(2C)}{\log n},$$

so we can get

$$\mathbb{E}\left[\left|\mathrm{supp}(M^{(L)})\right|\right] \le n^{1-\varepsilon_n}.$$

Our goal is to get $\mathbb{E}\left[\left|\mathrm{supp}(M^{(L)})\right|\right] \le n^{1-o(1)}$, so it suffices to show $\varepsilon_n = o(1)$. Therefore, we need to show that $\varepsilon_n > 0$ for all sufficiently large n and that $\lim_{n\to\infty}\varepsilon_n \to 0$.

Proof of $\varepsilon_n > 0$. First, since $T = \omega(\sqrt{\log\log n})$ (the assumption in the lemma statement), we have $T^2 = \omega(\log\log n)$. Therefore, we can get

$$\frac{T^2}{\sigma_M^2\log n} = \omega\left(\frac{\log\log n}{\log n}\right).$$

Also, we can see that $\frac{\log(2C)}{\log n} = o(1)$, and since $T^2 = \omega(\log\log n)$ whereas $\log(2C)$ and σ_M^2 are constant, we can get

$$\frac{\log(2C)}{\log n} = o\left(\frac{T^2}{\sigma_M^2\log n}\right).$$

Hence, we can get $\varepsilon_n > 0$ for all sufficiently large n.

Proof of $\lim_{n\to\infty}\varepsilon_n \to 0$. It suffices to show that when $n\to\infty$, we can get $\frac{T^2}{\sigma_M^2\log n}\to 0$, $\frac{\log\log n}{\log n}\to 0$, $\frac{\log(2C)}{\log n}\to 0$, respectively. Since $T = o(\sqrt{\log n})$, we have $T^2 = o(\log n)$, which directly implies

$$\frac{T^2}{\sigma_M^2\log n} \to 0.$$

Moreover, we can directly see

$$\frac{\log\log n}{\log n} \to 0$$

and

$$\frac{\log(2C)}{\log n} \to 0.$$

We have shown that $\varepsilon_n > 0$ eventually and $\varepsilon_n = o(1)$. Therefore,

$$\mathbb{E}\left[\left|\mathrm{supp}(M^{(L)})\right|\right] \le n^{1-\varepsilon_n} = n^{1-o(1)}.$$

This completes the proof. □

D.3 The Number of Non-Zero Entries

Next, we strengthen the result of Lemma D.2 by moving from an expectation bound to a high-probability guarantee. Specifically, we show that with high probability, the number of large entries in the query and key matrices is at most n^α for some $\alpha \in (0,1)$.

Lemma D.3 (The bound of the number of non-zero entries of $Q^{(L)}$ and $K^{(L)}$)**.** *Let the notation be defined as in Lemma D.2. For all $M^{(L)} \in \{Q^{(L)}, K^{(L)}, V^{(L)}\}$ and $\sigma_M \in \{\sigma_Q, \sigma_K, \sigma_V\}$, if $d = O(\log n)$, $T = o(\sqrt{\log n})$, and $T = \omega(\sqrt{\log\log n})$, then*

$$\Pr\left[\left|\mathrm{supp}\left(M^{(L)}\right)\right| > n^{1-o(1)}\right] \le \exp\left(-\Omega\left(n^{1-o(1)}\right)\right).$$

Proof. Since each $Q_{i,j} \in \mathrm{subG}(\sigma_Q^2)$, we have

$$\Pr\left[|Q_{i,j}| > T\right] \leq 2\exp\left(-\frac{T^2}{\sigma_Q^2}\right),$$

and similarly for $K_{i,j}$. Let $X = \left|\mathrm{supp}\left(Q^{(L)}\right)\right| = \sum_{i,j} \mathbb{I}\left[|Q_{i,j}| > T\right]$. We note that X is a sum of independent entries. Then, by Lemma D.2, we have

$$\mathbb{E}[X] \leq n^{1-o(1)}.$$

Therefore, by a multiplicative Chernoff bound (see Fact J.5), for all $\delta > 0$,

$$\Pr\left[X > (1+\delta)\mathbb{E}[X]\right] \leq \exp\left(-\frac{\delta^2 \cdot \mathbb{E}[X]}{3}\right).$$

In particular, for large enough n, $n^{1-o(1)} > (1+\delta)\mathbb{E}[X]$, so

$$\Pr\left[\left|\mathrm{supp}\left(Q^{(L)}\right)\right| > n^{1-o(1)}\right] \leq \exp\left(-\Omega(n^{1-o(1)})\right),$$

and likewise for $\left|\mathrm{supp}\left(K^{(L)}\right)\right|$. □

D.4 Approximating the Attention Computation Under the Sub-Gaussian Assumption

Finally, we combine the sparsity guarantees established in Lemma D.3 with our main result on attention approximation (Theorem C.3). The following theorem shows that, under the sub-Gaussian assumption, our algorithm approximates the attention computation and runs in sub-quadratic time with high probability. This provides a rigorous justification for the practical efficiency of our method and demonstrates that strong runtime guarantees can be achieved without requiring the bounded entry assumption, $B < o\left(\sqrt{\log n}\right)$.

Theorem D.4 (Main Theorem Under Sub-Gaussian Assumption)**.** *Let $Q, K, V \in \mathbb{R}^{n\times d}$ be the query, key, and value matrices. Suppose entries of Q, K are independent and sub-Gaussian with variance proxies σ_Q^2 and σ_K^2, respectively. Let $\epsilon \in (0, 0.1)$ be a target accuracy parameter. Then, with probability at least $1 - \exp(-\Omega(n^\alpha))$, for some $0 < \alpha < 1$, the output $P \in \mathbb{R}^{n\times d}$ from Algorithm 1 satisfies*

$$\left\|P - D^{-1}AV\right\|_\infty \leq \epsilon \cdot \|V\|_\infty,$$

where $A := \exp(QK^\top/d)$ and $D := \mathrm{diag}(A \cdot \mathbf{1}_n)$. Furthermore, the runtime of the algorithm is

$$O(n^{1+\alpha}).$$

Proof. By Lemma D.3, we have with probability $1 - \exp\left(-\Omega(n^\alpha)\right)$,

$$\left|\mathrm{supp}\left(M^{(L)}\right)\right| \leq n^\alpha,$$

with $\alpha \in (0, 1)$. Therefore, by Lemma B.9, we have

1. $\left|\mathrm{supp}\left(A^{(L)}\right)\right| = O\left(n^{1+\alpha}\right)$, and
2. it takes $O\left(n^{1+\alpha}d\right)$ time to compute $A^{(L)}$.

Since $\left|\mathrm{supp}\left(A^{(L)}\right)\right| = O\left(n^{1+\alpha}\right)$, computing $\left(\exp\left(A^{(L)}/d\right) - \mathbf{1}_{n\times n}\right)V$ (Line 6 Algorithm 3) and $\left(\exp\left(A^{(L)}\right) - \mathbf{1}_{n\times n}\right)\mathbf{1}_n$ (Line 7 in Algorithm 1) take $O\left(n^{1+\alpha}\right)$ time. □

E GENERALIZATION TO MULTI-LAYER TRANSFORMER ARCHITECTURES

Recall that from Definition 1.1 we define the single-layer attention computation as

$$D^{-1}AV = D^{-1}\exp\left(QK^\top/d\right)V.$$

In transformer architectures, this only denotes one single layer of the forward propagation. Without loss of generality, we can suppose this is the β-th layer: the query matrix is defined as $Q = X_\beta W_Q$, the key matrix is defined as $K = X_\beta W_K$, and the value matrix is defined as $V = X_\beta W_V$, where $X_\beta \in \mathbb{R}^{n\times d}$ is the input for the β-th layer and $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ are the weight matrices for the query, key, and value respectively. When attempting to optimize the β-th layer forward propagation, we can exactly compute the matrices Q, K, V in $O(nd^2)$ time, which is already sub-quadratic in n. The β-th layer can be formulated as the following recursive relation:

$$X_{\beta+1} \leftarrow D^{-1}\exp(X_\beta W_Q W_K^\top X_\beta^\top)X_\beta W_V$$

To generalize our result to multi-layer transformer architectures, we also need to efficiently approximate the gradients of the attention computation with respect to these weight matrices. We first define the attention computation loss function as follows:

Definition E.1 (Attention loss)**.** *Let $B \in \mathbb{R}^{n\times d}$ and $X, Y \in \mathbb{R}^{d\times d}$ be the weights where $X = W_Q W_K^\top$ and $Y = W_V$. Given the input $X_\ell \in \mathbb{R}^{n\times d}$, we define the attention loss as:*

$$\min_{X,Y\in\mathbb{R}^{d\times d}} L(X,Y) := \min_{X,Y\in\mathbb{R}^{d\times d}} \|D(X)^{-1}\exp(X_\ell X X_\ell^\top)X_\ell Y - B\|_F^2,$$

where the diagonal matrix $D(X) \in \mathbb{R}^{n\times n}$ is defined as $D(X) := \mathrm{diag}(\exp(X_\ell X X_\ell^\top)\mathbf{1}_n)$.

Since the dependence of the attention loss L on $Y = W_V$ is linear, it is very straightforward to compute the gradient with respect to the value weight $\frac{\partial L(X,Y)}{\partial Y}$. Therefore, prior works such as Alman and Song (2024b) focus on the optimization of $\frac{\partial L(X,Y)}{\partial X}$.

Definition E.2 (Approximate attention loss gradient computation)**.** *Let the attention loss $L(X,Y)$ be defined as in Definition E.1. For all $\epsilon > 0$, the goal is to output a vector $\widetilde{g}$ such that*

$$\left\|\widetilde{g} - \frac{\partial L(X,Y)}{\partial \mathrm{vec}\,(X)}\right\|_\infty \leq \epsilon.$$

With $x = \mathrm{vec}\,(X)$, $A\,(x)$ be the (Q,K)-softmax-attention matrix, and $\mathcal{Q}\,(x) := VV^\top A\,(x) - VE^\top$, Alman and Song (2024b); Deng et al. (2023a) show that the gradient of the attention can be expressed as

$$\frac{\partial L}{\partial x} = \mathrm{vec}\left(X_\ell\left(\mathcal{P}_1\,(x) - \mathcal{P}_2\,(x)\right)X_\ell^\top\right),$$

where $\mathcal{P}_1\,(x) = A\,(x)\circ\mathcal{Q}\,(x)$ and $\mathcal{P}_2\,(x) := A\,(x)\,\mathrm{diag}\,(r\,(x))$, with $r\,(x) \in \mathbb{R}^n$ being defined as $r\,(x)_i := \langle A\,(x)_{i,*}, \mathcal{Q}\,(x)_{i,*}\rangle$. Alman and Song (2024b) note that similar to the inference, the dominating term of the time complexity for computing $\frac{\mathrm{d}L}{\mathrm{d}x}$ is still the terms related to $A\,(x)$. Therefore, by replacing $A\,(x)$ with $U_1U_2^\top$ from Alman and Song (2023a), Alman and Song (2024b) shows that $\frac{\mathrm{d}L}{\mathrm{d}x}$ can be approximated in almost linear time.

Similarly, our method expresses $A\,(x)$ as $\exp\left(A^{(s)}\,(x)/d\right) + \exp\left(A^{(L)}\,(x)/d\right) - \mathbf{1}_{n\times n}$ (see Eq. (2)). Entries of $\exp\left(A^{(s)}/d\right)$ are small, so we can use the same technique as Alman and Song (2024b) to approximate it by $U_1U_2^\top$. $\exp\left(A^{(L)}\,(x)/d\right) - \mathbf{1}_{n\times n}$ is a sparse matrix, but the product with other dense matrices is dense. Thus, further computation may still require quadratic time.

To address this issue, we use the approximate singular value decomposition from Clarkson and Woodruff (2017). We can approximate its best rank-k low-rank approximation in $A^{(L)}\,(x)$ sparsity time (Theorem E.3). Choosing $k = n^{o(1)}$, we can approximate $\frac{\mathrm{d}L}{\mathrm{d}x}$ in $A^{(L)}\,(x)$ sparsity time, and by sub-Gaussianity, we can generalize our method to multi-layer attention.

Theorem E.3 (Theorem 8.1 in Clarkson and Woodruff (2017)). *For $A \in \mathbb{R}^{n\times n}$, there is an algorithm that, with failure probability $1/10$, finds matrices $L, W \in \mathbb{R}^{n\times k}$ with orthonormal columns, and diagonal $D \in \mathbb{R}^{k\times k}$, so that*

$$\|A - LDW^\top\|_F \leq (1+\varepsilon)\Delta_k.$$

The algorithm runs in time

$$O(\mathrm{nnz}(A)) + \widetilde{O}(nk^2\varepsilon^{-4} + k^3\varepsilon^{-5}).$$

F MULTIPLE THRESHOLDS SUPPORT BASIS DECOMPOSITION

We have now completed the presentation of the proofs for our first result, which uses the single-threshold support basis to approximate the attention computation without relying on the bounded-entry assumption required in Alman and Song (2023a). Instead, we introduce the sub-Gaussian assumption, which, to the best of our knowledge, is satisfied by all existing transformer architectures (see Appendix L).

To extend our theoretical results to settings where the sub-Gaussian assumption may not hold, we next present a framework based on multiple-threshold support basis decomposition. Unlike the single-threshold approach, this method does not require any distributional assumptions on the entries of Q and K. However, due to the lower bound established in Alman and Song (2023a) (see Theorem A.7), we must sacrifice the approximation error in order to achieve sub-quadratic runtime.

Beyond the optimization perspective offered by our multiple-threshold support basis decomposition, we also establish a theoretical connection between polynomial attention and the Chebyshev polynomial approximation of softmax attention (Definition 1.1). This provides a theoretical justification for the strong empirical performance of high-degree polynomial attention observed in Kacham et al. (2024).

Our theoretical results go further than merely explaining existing empirical findings. Specifically, we show that the sum of multiple polynomial attentions achieves a significantly smaller ℓ_p error with respect to softmax attention, even though the ℓ_∞ error remains the same as that of a single polynomial approximation. This suggests that aggregating multiple polynomial approximations more accurately captures the behavior of softmax attention. We hope this insight will inspire future empirical research to investigate the benefits of summing multiple polynomial attentions, rather than relying solely on a single polynomial approximation as in Kacham et al. (2024).

In Appendix F.1, we introduce some basic definitions for multiple thresholds support basis decomposition for decomposing the query Q and key K matrices. In Appendix F.2, we prove that the decomposition satisfies the definition of support basis. In Appendix F.3, we use the mathematical property of the support basis developed earlier to get the additive decomposition of the (Q,K)-softmax-attention matrix $\exp\left(QK^\top/d\right)$.

F.1 Basic Definitions

Now, we present the definitions for our multiple-threshold support basis decomposition. We aim to partition the query Q and key K matrices according to multiple value ranges, enabling a finer-grained approximation of the (Q,K)-softmax-attention matrix.

Definition F.1 (Multi-threshold decomposition of query and key matrices). *Let n, m, d be positive integers where $2 \leq m \leq n$. Let $Q, K \in \mathbb{R}^{n\times d}$ be the query and key matrices, respectively. Let $\min_{i\in[n],j\in[d]}\{|Q_{i,j}|, |K_{i,j}|\} = T_0 < T_1 < \cdots < T_{m-1} < T_m = \infty$ be a sequence of thresholds. For all $i \in [n]$ and $\ell \in [m]$, we define*

$$Q_{i,*}^{(T_\ell)} := \begin{cases} Q_{i,*} & \text{if } T_{\ell-1} \leq \max_{j\in[d]} |Q_{i,j}| < T_\ell \\ \mathbf{0}_{1\times d} & \text{otherwise,} \end{cases}$$

and similarly for $K_{i,j}^{(T_\ell)}$.

We now formally define how to decompose the matrix $QK^\top$ into a sum of components based on multiple thresholds. By ensuring that the resulting matrices are disjoint, we preserve the additive structure necessary for efficient and accurate approximation. The following definition specifies the multi-threshold disjoint decomposition of $QK^\top$.

Definition F.2 (Multi-threshold disjoint decomposition of $QK^\top$). *Let n, m, d be positive integers where $2 \leq m \leq n$. Let $Q, K \in \mathbb{R}^{n\times d}$ be the query and key matrices, respectively. Let $\min_{i\in[n],j\in[d]}\{|Q_{i,j}|, |K_{i,j}|\} = T_0 < T_1 < \cdots < T_{m-1} < T_m = \infty$ be a sequence of thresholds.*

For all $T_l, T_{l'} \in \{T_\ell\}_{\ell=1}^m$, we let $Q^{(T_l)}, K^{(T_{l'})}$ be defined as in Definition F.1, and we define

$$A^{(T_l,T_{l'})} := Q^{(T_l)} \left(K^{(T_{l'})}\right)^\top.$$

We now provide an example to better illustrate the definitions above.

Example F.3 (An example of 2-threshold disjoint decomposition of $QK^\top$)**.** *Let $T_1 > 0$ be a positive real number and $T_2 = \infty$. Let $\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \in \mathbb{R}^{3\times 2}$ and $\begin{bmatrix} g & i & k \\ h & j & l \end{bmatrix} \in \mathbb{R}^{2\times 3}$, where $|c|, |k| \geq T_1$ and the absolute value of other entries are smaller than T_1.*

This setup allows us to decompose the matrix product $QK^\top$ into a sum of disjoint components based on the threshold values T_1 and T_2 as follows:

$$\begin{aligned}
& \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \begin{bmatrix} g & i & k \\ h & j & l \end{bmatrix} \\
= & \begin{bmatrix} ag+bh & ai+bj & ak+bl \\ cg+dh & ci+dj & ck+dl \\ eg+hf & ei+fj & ek+fl \end{bmatrix} \\
= & \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & ck+dl \\ 0 & 0 & 0 \end{bmatrix}}_{A^{(T_2,T_2)}} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ cg+dh & ci+dj & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A^{(T_2,T_1)}} + \underbrace{\begin{bmatrix} 0 & 0 & ak+bl \\ 0 & 0 & 0 \\ 0 & 0 & ek+fl \end{bmatrix}}_{A^{(T_1,T_2)}} + \underbrace{\begin{bmatrix} ag+bh & ai+bj & 0 \\ 0 & 0 & 0 \\ eg+hf & ei+fj & 0 \end{bmatrix}}_{A^{(T_1,T_1)}} \\
= & \underbrace{\begin{bmatrix} 0 & 0 \\ c & d \\ 0 & 0 \end{bmatrix}}_{Q^{(T_2)}} \underbrace{\begin{bmatrix} 0 & 0 & k \\ 0 & 0 & l \end{bmatrix}}_{\left(K^{(T_2)}\right)^\top} + \underbrace{\begin{bmatrix} 0 & 0 \\ c & d \\ 0 & 0 \end{bmatrix}}_{Q^{(T_2)}} \underbrace{\begin{bmatrix} g & i & 0 \\ h & j & 0 \end{bmatrix}}_{\left(K^{(T_1)}\right)^\top} + \underbrace{\begin{bmatrix} a & b \\ 0 & 0 \\ e & f \end{bmatrix}}_{Q^{(T_1)}} \underbrace{\begin{bmatrix} 0 & 0 & k \\ 0 & 0 & l \end{bmatrix}}_{\left(K^{(T_2)}\right)^\top} + \underbrace{\begin{bmatrix} a & b \\ 0 & 0 \\ e & f \end{bmatrix}}_{Q^{(T_1)}} \underbrace{\begin{bmatrix} g & i & 0 \\ h & j & 0 \end{bmatrix}}_{\left(K^{(T_1)}\right)^\top}.
\end{aligned}$$

This example illustrates how the product $QK^\top$ can be partitioned into disjoint matrix components, forming a support basis that enables structured approximation.

F.2 Multi-Threshold Support Basis Decomposition of $QK^\top$

We now show that the multi-threshold decomposition of $QK^\top$ forms a valid support basis. This helps with applying polynomial approximations to each component individually.

Lemma F.4. *Let n, m, d be positive integers where $2 \leq m \leq n$. Let $Q, K \in \mathbb{R}^{n\times d}$ be the query and key matrices, respectively. Let $\min_{i\in[n],j\in[d]}\{|Q_{i,j}|, |K_{i,j}|\} = T_0 < T_1 < \cdots < T_{m-1} < T_m = \infty$ be a sequence of thresholds.*

Then, for all $T_l, T_{l'} \in \{T_\ell\}_{\ell=1}^m$, the collection $\{A^{(T_l,T_{l'})}\}_{T_l,T_{l'}\in\{T_\ell\}_{\ell=1}^m}$ forms a support basis (as defined in Definition B.5) of $QK^\top$.

Proof. By Definition B.5, it suffices to show:

1. $\sum_{l=1}^m \sum_{l'=1}^m A^{(T_l,T_{l'})} = QK^\top$ and
2. $A^{(T_l,T_{l'})}$'s are disjoint matrices (see Definition B.1).

Proof of Part 1.

By Definition F.1, we can see that

$$Q = \sum_{\ell=1}^m Q^{(T_\ell)} \quad \text{and} \quad K = \sum_{\ell=1}^m K^{(T_\ell)},$$

We have

$$\begin{aligned} QK^\top &= \left(\sum_{\ell=1}^m Q^{(T_\ell)}\right)\left(\sum_{\ell'=1}^m K^{(T_{\ell'})}\right)^\top \\ &= \sum_{\ell=1}^m \sum_{\ell'=1}^m Q^{(T_\ell)}\left(K^{(T_{\ell'})}\right)^\top \\ &= \sum_{\ell=1}^m \sum_{\ell'=1}^m A^{(T_\ell, T_{\ell'})}. \end{aligned}$$

Proof of Part 2.

Recall from Definition B.1, we say $A^{(T_l,T_{l'})}$'s are disjoint matrices if for all $i,j \in [n]$, for all $T_l, T_{l'} \in \{T_\ell\}_{\ell=1}^m$, for all $(\overline{T}_l, \overline{T}_{l'}) \in \{T_\ell\}_{\ell=1}^m \times \{T_\ell\}_{\ell=1}^m \setminus \{(T_l, T_{l'})\}$, if $\left(A^{(T_l,T_{l'})}\right)_{i,j} \neq 0$, then $\left(A^{(\overline{T}_l,\overline{T}_{l'})}\right)_{i,j} = 0$.

Suppose we have $\left(A^{(T_l,T_{l'})}\right)_{i,j} \neq 0$, which implies

$$\begin{aligned} \left(A^{(T_l,T_{l'})}\right)_{i,j} &= \left(Q^{(T_l)}\left(K^{(T_{l'})}\right)^\top\right)_{i,j} \\ &= Q_{i,*}^{(T_l)}\left(K_{j,*}^{(T_{l'})}\right)^\top \\ &\neq 0. \end{aligned}$$

This implies that $Q_{i,*}^{(T_l)} = Q_{i,*}$ and $K_{j,*}^{(T_l)} = K_{j,*}$.

Therefore, by Definition F.1, we can see that for all $\overline{T}_l \in \{T_\ell\}_{\ell=1}^m \setminus \{T_l\}$, for all $\overline{T}_{l'} \in \{T_\ell\}_{\ell=1}^m \setminus \{T_{l'}\}$, both $Q_{i,*}^{(\overline{T}_l)} = \mathbf{0}_{1\times d}$ and $K_{k,*}^{(\overline{T}_{l'})} = \mathbf{0}_{1\times d}$.

Finally, we can get for all $(\overline{T}_l, \overline{T}_{l'}) \in \{T_\ell\}_{\ell=1}^m \times \{T_\ell\}_{\ell=1}^m \setminus \{(T_l, T_{l'})\}$, $\left(A^{(\overline{T}_l,\overline{T}_{l'})}\right)_{i,j} = 0$. □

F.3 Additive Decomposition of $\exp\left(QK^\top/d\right)$

Applying the entry-wise exponential function to disjoint matrices allows us to express the exponential of their sum as the sum of their exponentials. We now generalize Fact B.2, originally stated for the single-threshold support basis, to the multi-threshold support basis setting as follows:

Lemma F.5. *Let n, m, d be positive integers where $2 \leq m \leq n$. Let $Q = \sum_{\ell=1}^m Q^{(T_\ell)} \in \mathbb{R}^{n\times d}$ and $K = \sum_{\ell'=1}^m K^{(T_{\ell'})} \in \mathbb{R}^{n\times d}$ be defined as in Definition F.1. Let $A^{(T_\ell,T_{\ell'})} = Q^{(T_\ell)}\left(K^{(T_{\ell'})}\right)^\top \in \mathbb{R}^{n\times n}$ be defined as in Definition F.2.*

Then, we have

$$\exp\left(QK^\top/d\right) = \sum_{\ell=1}^m \sum_{\ell'=1}^m \exp\left(A^{(T_\ell,T_{\ell'})}/d\right) - \left(m^2-1\right)\cdot \mathbf{1}_{n\times n}.$$

Proof. We prove this lemma using mathematical induction.

Base case (when $m = 2$).

We have

$$\begin{aligned} &\exp\left(QK^\top/d\right) \\ =\ &\exp\left(\left(Q^{(T_1)} + Q^{(T_2)}\right)\left(K^{(T_1)} + K^{(T_2)}\right)^\top/d\right) \\ =\ &\exp\left(\left(Q^{(T_1)}\left(K^{(T_1)}\right)^\top + Q^{(T_2)}\left(K^{(T_1)}\right)^\top + Q^{(T_1)}\left(K^{(T_2)}\right)^\top + Q^{(T_2)}\left(K^{(T_2)}\right)^\top\right)/d\right) \end{aligned}$$

$$
\begin{aligned}
&= \exp\left(\left(A^{(T_1,T_1)} + A^{(T_2,T_1)} + A^{(T_1,T_2)} + A^{(T_2,T_2)}\right)/d\right) \\
&= \exp\left(A^{(T_1,T_1)}/d\right) + \exp\left(A^{(T_2,T_1)}/d\right) + \exp\left(A^{(T_1,T_2)}/d\right) + \exp\left(A^{(T_2,T_2)}/d\right) - 3 \cdot \mathbf{1}_{n\times n} \\
&= \sum_{\ell=1}^{2}\sum_{\ell'=1}^{2} \exp\left(A^{(T_\ell,T_{\ell'})}/d\right) - (2^2-1)\cdot \mathbf{1}_{n\times n},
\end{aligned}
$$

where the first step follows from the definition of Q and K, the third step follows from the definition of $A^{(T_\ell,T_{\ell'})}$, the fourth step follows from combining Lemma F.4 and Fact B.2, and the last step follows from $m = 2$.

Inductive case.

Let t be an arbitrary positive integer greater than or equal to 2. Suppose for all $k \in [t]$, we have $Q = \sum_{\ell=1}^{t} Q^{(T_\ell)} \in \mathbb{R}^{n\times d}$ and $K = \sum_{\ell'=1}^{t} K^{(T_{\ell'})} \in \mathbb{R}^{n\times d}$, and we have

$$
\begin{aligned}
\exp\left(QK^\top/d\right) &= \exp\left(\sum_{\ell=1}^{t}\sum_{\ell'=1}^{t} A^{(T_\ell,T_{\ell'})}\right) \\
&= \sum_{\ell=1}^{t}\sum_{\ell'=1}^{t} \exp\left(A^{(T_\ell,T_{\ell'})}/d\right) - \left(t^2-1\right)\cdot \mathbf{1}_{n\times n}. \qquad (12)
\end{aligned}
$$

Now, we consider the case of $t+1$.

We have

$$
\begin{aligned}
&\exp\left(QK^\top/d\right) \\
&= \exp\left(\left(\sum_{\ell=1}^{t+1} Q^{(T_\ell)}\right)\left(\sum_{\ell'=1}^{t+1} K^{(T_{\ell'})}\right)^\top/d\right) \\
&= \exp\left(\left(\left(\sum_{\ell=1}^{t} Q^{(T_\ell)}\right)\left(\sum_{\ell'=1}^{t} K^{(T_{\ell'})}\right)^\top + \left(\sum_{\ell=1}^{t} Q^{(T_\ell)}\right)\left(K^{(T_{t+1})}\right)^\top\right.\right. \\
&\quad \left.\left. + \left(Q^{(T_{t+1})}\right)\left(\sum_{\ell'=1}^{t} K^{(T_{\ell'})}\right)^\top + \left(Q^{(T_{t+1})}\right)\left(K^{(T_{t+1})}\right)^\top\right)/d\right) \\
&= \exp\left(\left(\left(\sum_{\ell=1}^{t}\sum_{\ell'=1}^{t} A^{(T_\ell,T_{\ell'})}\right) + A^{(T_{t+1},T_{t+1})} + \left(\sum_{\ell'=1}^{t} A^{(T_{t+1},T_{\ell'})}\right) + \left(\sum_{\ell=1}^{t} A^{(T_\ell,T_{t+1})}\right)\right)/d\right) \\
&= \exp\left(\sum_{\ell=1}^{t}\sum_{\ell'=1}^{t} A^{(T_\ell,T_{\ell'})}/d\right) + \exp\left(A^{(T_{t+1},T_{t+1})}/d\right) + \exp\left(\sum_{\ell'=1}^{t} A^{(T_{t+1},T_{\ell'})}/d\right) \\
&\quad + \exp\left(\sum_{\ell=1}^{t} A^{(T_\ell,T_{t+1})}/d\right) - 3\cdot \mathbf{1}_{n\times n},
\end{aligned}
$$

where the first step follows from the definition of Q and K, the third step follows from the definition of $A^{(T_\ell,T_{\ell'})}$ (as defined in Definition F.2), and the fourth step follows from combining Lemma F.4 and Fact B.2.

We note that by Lemma F.4 and Fact B.2, we have

$$
\exp\left(\sum_{\ell'=1}^{t} A^{(T_{t+1},T_{\ell'})}\right) = \sum_{\ell'=1}^{t} \exp\left(A^{(T_{t+1},T_{\ell'})}\right) - (t-1)\cdot \mathbf{1}_{n\times n}
$$

and

$$
\exp\left(\sum_{\ell=1}^{t} A^{(T_\ell,T_{t+1})}\right) = \sum_{\ell=1}^{t} \exp\left(A^{(T_\ell,T_{t+1})}\right) - (t-1)\cdot \mathbf{1}_{n\times n}.
$$

Combining everything together, we have

$$\begin{aligned}\exp\left(QK^\top/d\right) &= \sum_{\ell=1}^{t+1}\sum_{\ell'=1}^{t+1}\exp\left(A^{(T_\ell,T_{\ell'})}/d\right) - \left(t^2-1+2\left(t-1\right)+3\right)\cdot \mathbf{1}_{n\times n}\\ &= \sum_{\ell=1}^{t+1}\sum_{\ell'=1}^{t+1}\exp\left(A^{(T_\ell,T_{\ell'})}/d\right) - \left((t+1)^2-1\right)\cdot \mathbf{1}_{n\times n},\end{aligned}$$

which completes the proof. □

G THRESHOLD VALUES SPECIFICATION AND INTERVAL-BASED FACTORIZATION

We have constructed a multiple-threshold support basis which can make the (Q,K)-softmax-attention matrix $\exp\left(QK^\top/d\right)$ be expressed as a sum of the exponential of each matrix in this support basis, namely $\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\exp\left(A^{(T_\ell,T_{\ell'})}/d\right) - \left(m^2-1\right)\cdot\mathbf{1}_{n\times n}$ (see Lemma F.5). We eventually want to approximate the attention computation problem $D^{-1}\exp\left(QK^\top/d\right)V$ (see Definition 1.1 for details). Therefore, we have to find a bound for m so that the total running time for approximating m^2 numbers of $D^{-1}\left(\exp\left(A^{(T_\ell,T_{\ell'})}/d\right) - \left(m^2-1\right)\cdot\mathbf{1}_{n\times n}\right)V$ can be upper bounded.

Specifically, in Appendix G.1, we specify the values of the thresholds $\{T_\ell\}_{\ell\in[m]\cup\{0\}}$ for our multiple-threshold support basis—motivated by Birge bucketing—so that m can eventually be bounded. In Appendix G.2, we demonstrate how to tightly bound $\|A^{\circ p}-B^{\circ p}\|_\infty$ using the Mean Value Theorem. In Appendix G.3, we show that the (Q,K)-softmax-attention matrix $\exp\left(QK^\top/d\right)$ (Definition 1.1) can be approximated by a sum of polynomial attention matrices (see Definition G.7) of the form $\left(U_1U_2^\top/d\right)^{\circ p}$. In Appendix G.4, we prove that the ℓ_∞ error between the (Q,K)-softmax-attention matrix and a single polynomial attention matrix is the same as that between the softmax matrix and the sum of polynomial attention matrices. Finally, in Appendix G.5, we show that the ℓ_1 error is significantly reduced when using Birge-bucketing-inspired thresholds.

G.1 Thresholds for Support Basis

Birge bucketing (Birgé, 1987; Batu et al., 2004; Canonne et al., 2016; Batu et al., 2000; Aliakbarpour et al., 2023) is a well-known technique used in distribution testing: it partitions the support of a distribution into buckets to better analyze its tail behavior and test for heavy-tailedness. In particular, Aliakbarpour et al. (2023) introduces an equal-weight bucketing scheme that partitions the domain of a distribution into intervals (or buckets) such that each bucket contains the same probability mass.

Unlike the single-threshold support basis, which assumes that the entries of $Q,K\in\mathbb{R}^{n\times d}$ are sub-Gaussian, the multiple-threshold support basis makes no specific distributional assumptions about the entries of Q and K. However, since their entries still follow some underlying distribution, we can apply the Birge bucketing technique to partition them. Because the exact distribution of the entries of Q and K is unknown[3], we cannot perform weight-based bucketing as in Aliakbarpour et al. (2023). Moreover, uniform bucketing—i.e., using a fixed bucket length of $o\left(\sqrt{\log n}\right)$—is also impractical, as the entries may take extremely large values, leading to an excessively large number of buckets m. Since we must approximate $\exp\left(A^{(T_\ell,T_{\ell'})}/d\right) - \left(m^2-1\right)\cdot\mathbf{1}_{n\times n}$ for all $(\ell,\ell')\in[m]\times[m]$, a large value of m would prevent sub-quadratic runtime.

To keep m small, we define the thresholds as $T_\ell = b\left(1+\epsilon\right)^\ell$. We formally present the following definition:

Definition G.1. *Let $\epsilon<0$ be the bucketing parameter. Let n,m,d be positive integers where $2\le m\le n$. Let $Q,K\in\mathbb{R}^{n\times d}$ be the query and key matrices, respectively. Let $b=\min_{i\in[n],j\in[d]}\{|Q_{i,j}|,|K_{i,j}|\}$ and $B=\max_{i\in[n],j\in[d]}\{|Q_{i,j}|,|K_{i,j}|\}$. For all $\ell\in[m]\cup\{0\}$, we let $T_\ell = b\left(1+\epsilon\right)^\ell$.*

We define

$$Q_{i,*}^{(T_\ell)} := \begin{cases} Q_{i,*} & \text{if } T_{\ell-1}\le \max_{j\in[d]}|Q_{i,j}| < T_\ell \\ \mathbf{0}_{1\times d} & \text{otherwise,}\end{cases}$$

[3]We even suppose that we do not have sample access to the entries of Q and K.

and similarly for $K_{i,j}^{(T_\ell)}$.

Using $T_\ell = b\,(1+\epsilon)^\ell$, we can eventually show that the number of buckets m can be bounded. It suffices to have $\lfloor \log_{1+\epsilon}(B/b) \rfloor + 1$ numbers of buckets.

Fact G.2. *Let $\epsilon < 0$ be the bucketing parameter. Let n, m, d be positive integers where $2 \leq m \leq n$. Let $Q, K \in \mathbb{R}^{n \times d}$ be the query and key matrices, respectively. Let $b = \min_{i\in[n], j\in[d]} \{|Q_{i,j}|, |K_{i,j}|\}$ and $B = \max_{i\in[n], j\in[d]} \{|Q_{i,j}|, |K_{i,j}|\}$. For all $\ell \in [m] \cup \{0\}$, we let $T_\ell = b\,(1+\epsilon)^\ell$. Let*

$$M = \left\{ m_i \mid [b, B] \subseteq \bigcup_{\ell=0}^{m_i - 1} [T_\ell, T_{\ell+1}) \right\}$$

be a sequence of positive integers.

Then, we have

$$\min_i M = \lfloor \log_{1+\epsilon}(B/b) \rfloor + 1.$$

Proof. We note that $\bigcup_{\ell=0}^{m-1} [T_\ell, T_{\ell+1})$ is an interval and, by definition, $T_0 = b\,(1+\epsilon)^0 = b$. Therefore, it suffices to find the smallest m such that

$$T_m > B.$$

We have

$$\begin{aligned} T_m = b\,(1+\epsilon)^m &> B \\ (1+\epsilon)^m &> B/b \\ m &> \log_{1+\epsilon}(B/b). \end{aligned}$$

Therefore, the smallest possible m is $\lfloor \log_{1+\epsilon}(B/b) \rfloor + 1$. □

We need to ensure that within each bucket $(\ell, \ell') \in [m] \times [m]$, we can make $\|Q^{(T_\ell)}\|_\infty, \|K^{(T_\ell)}\|_\infty < o\left(\sqrt{\log n}\right)$. Although $\|Q^{(T_\ell)}\|_\infty, \|K^{(T_\ell)}\|_\infty \in \mathbb{R}$, without imposing any assumption, they can be much larger than $o\left(\sqrt{\log n}\right)$. Therefore, we take out a large constant $C^{(T_\ell)}$ from $Q^{(T_\ell)}$ and $K^{(T_{\ell'})}$ to make $Q^{(T_\ell)} = C^{(T_\ell)} Q^{(\ell)}$ and $K^{(T_{\ell'})} = C^{(T_{\ell'})} K^{(\ell')}$, where $\|Q^{(\ell)}\|_\infty, \left\|K^{(\ell')}\right\|_\infty < o\left(\sqrt{\log n}\right)$. Furthermore, we show that $C^{(T_\ell, T_{\ell'})} = C^{(T_\ell)} C^{(T_{\ell'})} \geq \frac{b^2 (1+\epsilon)^{\ell+\ell'}}{\log n}$, where $b = \min_{i\in[n], j\in[d]} \{|Q_{i,j}|, |K_{i,j}|\}$ and $\epsilon < 0$ is the bucketing parameter.

Lemma G.3 (Normalized block decomposition)**.** *Let n, m, d be positive integers where $2 \leq m \leq n$. Let $\epsilon < 0$ be the bucketing parameter. Let $Q = \sum_{\ell=1}^m Q^{(T_\ell)} \in \mathbb{R}^{n\times d}$ and $K = \sum_{\ell'=1}^m K^{(T_{\ell'})} \in \mathbb{R}^{n\times d}$ be defined as in Definition G.1, with $b = \min_{i\in[n], j\in[d]} \{|Q_{i,j}|, |K_{i,j}|\}$, $B = \max_{i\in[n], j\in[d]} \{|Q_{i,j}|, |K_{i,j}|\}$, and for all $\ell \in [m] \cup \{0\}$, we let $T_\ell = b\,(1+\epsilon)^\ell$. Let $A^{(T_\ell, T_{\ell'})} = Q^{(T_\ell)} \left(K^{(T_{\ell'})}\right)^\top \in \mathbb{R}^{n\times n}$.*

Let $m := \lfloor \log_{1+\epsilon}(B/b) \rfloor + 1$ and $C^{(T_\ell, T_{\ell'})} \geq \frac{b^2(1+\epsilon)^{\ell+\ell'}}{\log n}$. By Lemma F.5, we have

$$\exp\left(QK^\top\right) = \sum_{\ell=1}^m \sum_{\ell'=1}^m \exp\left(A^{(T_\ell, T_{\ell'})}\right) - \left(m^2 - 1\right) \cdot \mathbf{1}_{n\times n}.$$

Then, for all $A^{(T_\ell, T_{\ell'})}$, there exists $Q^{(\ell)}, K^{(\ell')} \in \mathbb{R}^{n\times d}$ such that

$$A^{(T_\ell, T_{\ell'})} = C^{(T_\ell, T_{\ell'})} \cdot \underbrace{Q^{(\ell)} \left(K^{(\ell')}\right)^\top}_{:= A^{(\ell, \ell')}},$$

and $\|Q^{(\ell)}\|_\infty, \left\|K^{(\ell')}\right\|_\infty \leq o\left(\sqrt{\log n}\right)$.

Proof. By the definition of $A^{(T_\ell, T_{\ell'})}$, we have for all $p, q \in [n]$,

$$\left(A^{(T_\ell, T_{\ell'})}\right)_{p,q} = \sum_{k=1}^{d} \left(Q^{(T_\ell)}\right)_{p,k} \left(K^{(T_{\ell'})}\right)_{q,k}.$$

By Definition G.1, we notice that $\left\|Q^{(T_\ell)}\right\|_\infty \leq b\left(1+\epsilon\right)^\ell$ and $\left\|K^{\left(T'_\ell\right)}\right\|_\infty \leq b\left(1+\epsilon\right)^{\ell'}$.

By letting $C^{(T_\ell)} = \frac{b(1+\epsilon)^\ell}{\sqrt{\log n}}$, $K^{(T_{\ell'})} = C^{(T_{\ell'})} K^{\left(\ell'\right)}$, and $Q^{(T_\ell)} = C^{(T_\ell)} Q^{(\ell)}$, we have

$$\left\|Q^{(\ell)}\right\|_\infty, \left\|K^{\left(\ell'\right)}\right\|_\infty \leq o\left(\sqrt{\log n}\right).$$

Additionally, we have

$$\begin{aligned}
\left(A^{(T_\ell, T_{\ell'})}\right)_{p,q} &= \sum_{k=1}^{d} \left(Q^{(T_\ell)}\right)_{p,k} \left(K^{(T_{\ell'})}\right)_{q,k} \\
&= \sum_{k=1}^{d} \left(C^{(T_\ell)} Q^{(\ell)}\right)_{p,k} \left(C^{(T_{\ell'})} K^{\left(\ell'\right)}\right)_{q,k} \\
&= C^{(T_\ell)} C^{(T_{\ell'})} \sum_{k=1}^{d} \left(Q^{(\ell)}\right)_{p,k} \left(K^{\left(\ell'\right)}\right)_{q,k} \\
&= C^{(T_\ell)} C^{(T_{\ell'})} \cdot Q^{(\ell)} \left(K^{\left(\ell'\right)}\right)^\top.
\end{aligned}$$

Defining $C^{(T_\ell, T_{\ell'})} := C^{(T_\ell)} C^{(T_{\ell'})}$, we have

$$\begin{aligned}
C^{(T_\ell)} C^{(T_{\ell'})} &= \frac{b\left(1+\epsilon\right)^\ell}{\sqrt{\log n}} \frac{b\left(1+\epsilon\right)^{\ell'}}{\sqrt{\log n}} \\
&= \frac{b^2\left(1+\epsilon\right)^{\ell+\ell'}}{\log n}.
\end{aligned}$$

□

G.2 Basic Facts

Now, we use the Mean Value Theorem to show that for all arbitrary matrices A, B, positive integers p, $\beta = \max\left\{\|A\|_\infty, \|B\|_\infty\right\}$, and $\epsilon > 0$, we can get $\|A^{\circ p} - B^{\circ p}\|_\infty \leq p \cdot \beta^{p-1} \cdot \epsilon$.

Fact G.4. *Let $A, B \in \mathbb{R}^{n \times n}$, and let $p \in \mathbb{N}$ with $p \geq 1$. Let $\beta := \max\left\{\|A\|_\infty, \|B\|_\infty\right\}$. Suppose $\|A - B\|_\infty < \epsilon$. Then, we have:*

$$\|A^{\circ p} - B^{\circ p}\|_\infty \leq p \cdot \beta^{p-1} \cdot \epsilon.$$

Proof. Let $f\left(x\right) = x^p$, which is differentiable on $\mathbb{R}$. By the Mean Value Theorem, there exists a point c between a and b such that:

$$f\left(a\right) - f\left(b\right) = f'\left(c\right)\left(a - b\right).$$

Since $f'\left(x\right) = px^{p-1}$, we obtain:

$$a^p - b^p = pc^{p-1}\left(a - b\right).$$

Taking absolute values on both sides gives:

$$|a^p - b^p| = |pc^{p-1}||a - b| = p|c|^{p-1}|a - b|.$$

Because c lies between a and b, we have $|c| \leq \max\left\{|a|, |b|\right\}$. Therefore, we have

$$|a^p - b^p| \leq p \cdot \max\left\{|a|, |b|\right\}^{p-1} \cdot |a - b| < p \cdot \max\left\{|a|, |b|\right\}^{p-1} \cdot \epsilon.$$

□

Remark G.5. *By Lemma F.5, we have*

$$\exp\left(QK^\top/d\right) = \sum_{\ell=1}^m \sum_{\ell'=1}^m \exp\left(A^{(T_\ell,T_{\ell'})}/d\right) - \left(m^2-1\right)\cdot \mathbf{1}_{n\times n}.$$

With some abuse of notation, we may say

$$\exp\left(QK^\top/d\right) = \sum_{\ell=1}^m \sum_{\ell'=1}^m \exp\left(A^{(T_\ell,T_{\ell'})}/d\right).$$

This does not affect the correctness and running time of our algorithm. We note that since the collection $\{A^{(T_l,T_{l'})}\}_{T_l,T_{l'}\in\{T_\ell\}_{\ell=1}^m}$ forms a support basis (as defined in Definition B.5) of $QK^\top$, $\exp\left(A^{(T_\ell,T_{\ell'})}/d\right)$ transforms the zero entries to $\exp(0)=1$. Subtracting $\left(m^2-1\right)\cdot\mathbf{1}_{n\times n}$ is equivalent to say that we redefine "$\exp(0)=1$" to "$\exp(0)=0$" in $\sum_{\ell=1}^m\sum_{\ell'=1}^m \exp\left(A^{(T_\ell,T_{\ell'})}/d\right)$.

We have shown that $Q^{(T_\ell)} = C^{(T_\ell)}Q^{(\ell)}$ and $K^{(T_{\ell'})} = C^{(T_{\ell'})}K^{(\ell')}$ in Lemma G.3. Since in this paper, we apply $\exp(A)$ entry-wisely to a matrix, we can get for all $c\in\mathbb{R}$, $\exp(cA) = \exp(A)^{\circ c}$.

Fact G.6. *Let n,m,d be positive integers where $2\le m\le n$. Let $\epsilon\in(0,1)$. Let $Q=\sum_{\ell=1}^m Q^{(T_\ell)}\in\mathbb{R}^{n\times d}$ and $K=\sum_{\ell'=1}^m K^{(T_{\ell'})}\in\mathbb{R}^{n\times d}$ be defined as in Definition G.1, with $b=\min_{i\in[n],j\in[d]}\{|Q_{i,j}|,|K_{i,j}|\}$, $B=\max_{i\in[n],j\in[d]}\{|Q_{i,j}|,|K_{i,j}|\}$, and for all $\ell\in[m]\cup\{0\}$, we let $T_\ell = b\left(1+\epsilon\right)^\ell$. Let $A^{(T_\ell,T_{\ell'})} = Q^{(T_\ell)}\left(K^{(T_{\ell'})}\right)^\top = C^{(T_\ell,T_{\ell'})}\cdot Q^{(\ell)}\left(K^{(\ell')}\right)^\top\in\mathbb{R}^{n\times n}$. Let $m := \lfloor\log_{1+\epsilon}(B/b)\rfloor+1$ and $C^{(T_\ell,T_{\ell'})} = \frac{b^2(1+\epsilon)^{\ell+\ell'}}{\log n}$.*

Then, we have

- **Part 1.**

$$\exp\left(QK^\top/d\right) = \sum_{\ell=1}^m\sum_{\ell'=1}^m \exp\left(Q^{(\ell)}\left(K^{(\ell')}\right)^\top/d\right)^{\circ C^{(T_\ell,T_{\ell'})}},$$

- **Part 2.** *and for all $(\ell,\ell')\in[m]\times[m]$,*

$$\left\|Q^{(\ell)}\left(K^{(\ell')}\right)^\top/d\right\|_\infty \le o\left(\log n\right)$$

Proof. **Proof of Part 1.**

We have

$$\begin{aligned}
\exp\left(QK^\top/d\right) =& \sum_{\ell=1}^m\sum_{\ell'=1}^m \exp\left(A^{(T_\ell,T_{\ell'})}/d\right)\\
=& \sum_{\ell=1}^m\sum_{\ell'=1}^m \exp\left(C^{(T_\ell,T_{\ell'})}\cdot Q^{(\ell)}\left(K^{(\ell')}\right)^\top/d\right)\\
=& \sum_{\ell=1}^m\sum_{\ell'=1}^m \exp\left(Q^{(\ell)}\left(K^{(\ell')}\right)^\top/d\right)^{\circ C^{(T_\ell,T_{\ell'})}},
\end{aligned}$$

where the first step follows from Remark G.5, the second step follows from the lemma statement, and the last step follows from $\exp(cA)=\exp(A)^{\circ c}$ with the degree c is applied entry-wisely to the matrix $\exp(A)$.

Proof of Part 2.

Similar as Lemma B.8, for all arbitrary $(\ell,\ell')\in[m]\times[m]$, we have

$$\left\|Q^{(\ell)}\left(K^{(\ell')}\right)^\top\right\|_\infty = \max_{(i,j)\in[n]\times[n]}\left|\left(Q^{(\ell)}\left(K^{(\ell')}\right)^\top\right)_{i,j}\right|$$

$$
\begin{aligned}
&= \max_{(i,j)\in[n]\times[n]} \left| \sum_{l=1}^{d} Q_{i,l}^{(\ell)} K_{j,l}^{(\ell')} \right| \\
&\leq \sum_{l=1}^{d} \left| \max_{i\in[n]} Q_{i,l}^{(\ell)} \right| \cdot \left| \max_{j\in[n]} K_{j,l}^{(\ell')} \right| \\
&\leq \sum_{l=1}^{d} o\left(\sqrt{\log n}\right)^2 \\
&= o\left(d \log n\right),
\end{aligned}
$$

where the first step follows from the definition of the ℓ_∞ norm, the third step follows from the triangle inequality (see Fact J.3), and the fourth step follows from Lemma G.3. □

G.3 Reducing the Softmax Attention Matrix to the Polynomial Attention Matrix

We define the polynomial attention matrix, which replaces the exponential in the standard attention formula with a polynomial of a given degree.

Definition G.7 (Polynomial attention matrix (Kacham et al., 2024))**.** *Let p be an arbitrary positive integer. Let $Q, K \in \mathbb{R}^{n\times d}$. The polynomial attention matrix $A \in \mathbb{R}^{n\times n}$ is defined as $A := \left(QK^\top\right)^{\circ p}$.*

Now, we show that the (Q, K)-softmax-attention matrix is "close" to the sum of polynomial attention matrices.

Lemma G.8. *Let n, m, d be positive integers where $2 \leq m \leq n$. Let $\epsilon \in (0, 1)$. Let $Q = \sum_{\ell=1}^{m} Q^{(T_\ell)} \in \mathbb{R}^{n\times d}$ and $K = \sum_{\ell'=1}^{m} K^{(T_{\ell'})} \in \mathbb{R}^{n\times d}$ be defined as in Definition G.1, with $b = \min_{i\in[n],j\in[d]} \{|Q_{i,j}|, |K_{i,j}|\}$, $B = \max_{i\in[n],j\in[d]} \{|Q_{i,j}|, |K_{i,j}|\}$, and for all $\ell \in [m] \cup \{0\}$, we let $T_\ell = b\left(1+\epsilon\right)^\ell$. Let $A^{(T_\ell, T_{\ell'})} = Q^{(T_\ell)} \left(K^{(T_{\ell'})}\right)^\top = C^{(T_\ell, T_{\ell'})} \cdot Q^{(\ell)} \left(K^{(\ell')}\right)^\top \in \mathbb{R}^{n\times n}$.*

Let $m := \left\lfloor \log_{1+\epsilon}(B/b) \right\rfloor + 1$ and $C^{(T_\ell, T_{\ell'})} = \frac{b^2(1+\epsilon)^{\ell+\ell'}}{\log n}$. Let $\beta = \max\left\{\|Q\|_\infty, \|K\|_\infty\right\}$.

Suppose by Lemma A.4, there exists $U_1^{(\ell,\ell')}, U_2^{(\ell,\ell')} \in \mathbb{R}^{n\times r}$ such that $U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top$ is the (ϵ_0, r)-approximation of $\exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top\right)$.

Then, we have

$$
\left\| \exp\left(QK^\top/d\right) - \sum_{\ell=1}^{m}\sum_{\ell'=1}^{m} \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right)^{\circ C^{(T_\ell, T_{\ell'})}} \right\|_\infty \leq O\left(\frac{B^2}{\log n} e^{o(B^2)} \epsilon_0 \right).
$$

Proof. By **Part 1** of Fact G.6, we have

$$
\exp\left(QK^\top/d\right) = \sum_{\ell=1}^{m}\sum_{\ell'=1}^{m} \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top /d \right)^{\circ C^{(T_\ell, T_{\ell'})}}.
$$

By Lemma A.4, since for all $(\ell, \ell') \in [m] \times [m]$, we have $\left\| Q^{(\ell)} \left(K^{(\ell')}\right)^\top /d \right\|_\infty \leq o\left(\log n\right)$ (see **Part 2** of Fact G.6), we can construct $U_1^{(\ell,\ell')}, U_2^{(\ell,\ell')} \in \mathbb{R}^{n\times r}$ such that

$$
\left\| \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top /d \right) - U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right\|_\infty < \epsilon_0. \tag{13}
$$

By the triangle inequality (see Fact J.3), we have

$$\begin{aligned}
&\left\| \sum_{\ell=1}^m \sum_{\ell'=1}^m \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right)^{\circ C^{(T_\ell, T_{\ell'})}} - \sum_{\ell=1}^m \sum_{\ell'=1}^m \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right)^{\circ C^{(T_\ell, T_{\ell'})}} \right\|_\infty \\
\leq & \sum_{\ell=1}^m \sum_{\ell'=1}^m \left\| \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right)^{\circ C^{(T_\ell, T_{\ell'})}} - \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right)^{\circ C^{(T_\ell, T_{\ell'})}} \right\|_\infty \\
\leq & \sum_{\ell=1}^m \sum_{\ell'=1}^m C^{(T_\ell, T_{\ell'})} \beta^{C^{(T_\ell, T_{\ell'})} - 1} \left\| \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right) - U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right\|_\infty, \qquad (14)
\end{aligned}$$

where the second step follows from Fact G.4 with

$$\beta = \max\left\{ \left\| \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right) \right\|_\infty, \left\| U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right\|_\infty \right\}.$$

Considering $C^{(T_\ell, T_{\ell'})}$, we have

$$\begin{aligned}
\max_{T_\ell, T_{\ell'}} C^{(T_\ell, T_{\ell'})} &= C^{(T_m, T_m)} \\
&= \frac{b^2 (1+\epsilon)^{2m}}{\log n} \\
&= \frac{B^2}{\log n},
\end{aligned}$$

where the first and the second step follows from the fact that $C^{(T_\ell, T_{\ell'})} = \frac{b^2 (1+\epsilon)^{\ell+\ell'}}{\log n}$ is strictly increasing (see from the lemma statement), the third step follows from $B = b(1+\epsilon)^m$.

Considering β, by Fact G.4, we have

$$\begin{aligned}
\beta = & \max\left\{ \left\| \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right) \right\|_\infty, \left\| U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right\|_\infty \right\} \\
\leq & \left\| \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right) \right\|_\infty + \epsilon_0 \\
= & \exp\left(\left\| Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right\|_\infty \right) + \epsilon_0 \\
= & \exp\left(o(\log n) \right) + \epsilon_0 \\
= & n^{o(1)},
\end{aligned}$$

where the first step follows from Fact G.4, the second step follows from Eq. (13), and the fourth step follows from **Part 2** of Fact G.6.

Combining everything together, we have

$$\begin{aligned}
&\left\| \sum_{\ell=1}^m \sum_{\ell'=1}^m \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right)^{\circ C^{(T_\ell, T_{\ell'})}} - \sum_{\ell=1}^m \sum_{\ell'=1}^m \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right)^{\circ C^{(T_\ell, T_{\ell'})}} \right\|_\infty \\
\leq & O\left(\frac{B^2}{\log n} n^{\frac{o(1) \cdot B^2}{\log n}} \epsilon_0 \right)
\end{aligned}$$

□

G.4 Reducing to Polynomial Attention Without Bucketing

Additionally, we note that the bucketing technique does not improve the ℓ_∞-closeness between the polynomial attention matrix and the (Q,K)-softmax attention matrix. Without using this bucketing technique, we obtain only a single polynomial attention matrix, which is precisely the case studied in Kacham et al. (2024).

Lemma G.9. *Let n, m, d be positive integers. Let $\epsilon > 0$. Let $Q = \sum_{\ell=1}^{m} Q^{(T_\ell)} \in \mathbb{R}^{n\times d}$ and $K = \sum_{\ell'=1}^{m} K^{(T_{\ell'})} \in \mathbb{R}^{n\times d}$ be defined as in Definition G.1, with $b = \min_{i\in[n],j\in[d]} \{|Q_{i,j}|, |K_{i,j}|\}$, $B = \max_{i\in[n],j\in[d]} \{|Q_{i,j}|, |K_{i,j}|\}$, and for all $\ell \in [m]\cup\{0\}$, we let $T_\ell = b(1+\epsilon)^\ell$. Let $A^{(T_\ell,T_{\ell'})} = Q^{(T_\ell)} \left(K^{(T_{\ell'})}\right)^\top = C^{(T_\ell,T_{\ell'})} \cdot Q^{(\ell)} \left(K^{(\ell')}\right)^\top \in \mathbb{R}^{n\times n}$. Let $\beta = \max\{\|Q\|_\infty, \|K\|_\infty\}$.*

Suppose $m = 1$ and by Lemma A.4, there exists $U_1^{(1,1)}, U_2^{(1,1)} \in \mathbb{R}^{n\times r}$ such that $U_1^{(1,1)} \left(U_2^{(1,1)}\right)^\top$ is the (ϵ_0, r)-approximation of $\exp\left(Q^{(1)} \left(K^{(1)}\right)^\top / d\right)$.

Then, we have

$$\left\| \exp\left(QK^\top/d\right) - \left(U_1^{(1,1)} \left(U_2^{(1,1)}\right)^\top\right)^{\circ C^{(T_1,T_1)}} \right\|_\infty \leq O\left(\frac{B^2}{\log n} e^{o(B^2)} \epsilon_0\right).$$

Proof. This proof is similar to that of Lemma G.8. By **Part 1** of Fact G.6, with $m = 1$, we have

$$\exp\left(QK^\top/d\right) = \exp\left(Q^{(1)} \left(K^{(1)}\right)^\top / d\right)^{\circ C^{(T_1,T_1)}},$$

with

$$\begin{aligned} C^{(T_1,T_1)} &= \frac{b^2 (1+\epsilon)^{1+1}}{\log n} \\ &= \frac{b^2 (1+\epsilon)^{2m}}{\log n} \\ &= B^2 / \log n, \end{aligned}$$

where the first step follows from the definition of $C^{(T_\ell,T_{\ell'})}$, the second step follows from $m = 1$, and the last step follows from $B = b(1+\epsilon)^m$.

By Lemma A.4, since we have $\left\|Q^{(1)} \left(K^{(1)}\right)^\top / d\right\|_\infty \leq o(\log n)$ (see **Part 2** of Fact G.6), we can construct $U_1^{(1,1)}, U_2^{(1,1)} \in \mathbb{R}^{n\times r}$ such that

$$\left\| \exp\left(Q^{(1)} \left(K^{(1)}\right)^\top / d\right) - U_1^{(1,1)} \left(U_2^{(1,1)}\right)^\top \right\|_\infty < \epsilon_0. \tag{15}$$

We have

$$\begin{aligned} &\left\| \exp\left(Q^{(1)} \left(K^{(1)}\right)^\top / d\right)^{\circ C^{(T_1,T_1)}} - \left(U_1^{(1,1)} \left(U_2^{(1,1)}\right)^\top\right)^{\circ C^{(T_1,T_1)}} \right\|_\infty \\ \leq\ & C^{(T_1,T_1)} \beta^{C^{(T_1,T_1)}-1} \left\| \exp\left(Q^{(1)} \left(K^{(1)}\right)^\top / d\right) - U_1^{(1,1)} \left(U_2^{(1,1)}\right)^\top \right\|_\infty \\ =\ & C^{(T_1,T_1)} \beta^{C^{(T_1,T_1)}-1} \epsilon_0, \end{aligned}$$

where the first step follows from Fact G.4 and the second step follows from Eq. (15).

Considering β, we have

$$\beta = \max\left\{ \left\| \exp\left(Q^{(1)} \left(K^{(1)}\right)^\top / d\right) \right\|_\infty, \left\| U_1^{(1,1)} \left(U_2^{(1,1)}\right)^\top \right\|_\infty \right\}$$

$$
\begin{aligned}
&\leq \left\| \exp\left(Q^{(1)} \left(K^{(1)}\right)^\top / d \right) \right\|_\infty + \epsilon_0 \\
&= \exp\left(\left\| Q^{(1)} \left(K^{(1)}\right)^\top / d \right\|_\infty \right) + \epsilon_0 \\
&= \exp\left(o\left(\log n\right) \right) + \epsilon_0 \\
&= n^{o(1)},
\end{aligned}
$$

where the first step follows from Fact G.4, the second step follows from Eq. (15), and the fourth step follows from **Part 2** of Fact G.6.

Combining everything together, we have

$$
\left\| \exp\left(Q^{(1)} \left(K^{(1)}\right)^\top / d \right)^{\circ C^{(T_1,T_1)}} - \left(U_1^{(1,1)} \left(U_2^{(1,1)}\right)^\top \right)^{\circ C^{(T_1,T_1)}} \right\|_\infty \leq O\left(\frac{B^2}{\log n} n^{\frac{o(1)\cdot B^2}{\log n}} \epsilon_0 \right).
$$

□

G.5 Bucketing Reduces the Error in ℓ_1 Norm

In the following lemma, we analyze how the ℓ_1 error between the (Q,K)-softmax-attention matrix and its polynomial approximation is affected by the bucketing technique. While prior sections mainly focus on ℓ_∞ approximations, the ℓ_1 norm offers a complementary view by aggregating errors across entries. We show that the sum of multiple polynomial attention matrices generated by the bucketing technique significantly reduces the ℓ_1 error, thereby providing a stronger global approximation guarantee.

First, we analyze the case where we use the bucketing technique.

Lemma G.10. *Let n, m, d be positive integers where $2 \leq m \leq n$. Let $\epsilon \in (0,1)$. Let $Q = \sum_{\ell=1}^m Q^{(T_\ell)} \in \mathbb{R}^{n\times d}$ and $K = \sum_{\ell'=1}^m K^{(T_{\ell'})} \in \mathbb{R}^{n\times d}$ be defined as in Definition G.1, with $b = \min_{i\in[n],j\in[d]} \{|Q_{i,j}|, |K_{i,j}|\}$, $B = \max_{i\in[n],j\in[d]} \{|Q_{i,j}|, |K_{i,j}|\}$, and for all $\ell \in [m]\cup\{0\}$, we let $T_\ell = b\left(1+\epsilon\right)^\ell$. Let $A^{(T_\ell,T_{\ell'})} = Q^{(T_\ell)} \left(K^{(T_{\ell'})}\right)^\top = C^{(T_\ell,T_{\ell'})} \cdot Q^{(\ell)} \left(K^{(\ell')}\right)^\top \in \mathbb{R}^{n\times n}$.*

Let $m := \left\lfloor \log_{1+\epsilon}\left(B/b\right) \right\rfloor + 1$ and $C^{(T_\ell,T_{\ell'})} = \frac{b^2(1+\epsilon)^{\ell+\ell'}}{\log n}$. Let $\beta = \max\left\{ \|Q\|_\infty, \|K\|_\infty \right\}$.

Suppose by Lemma A.4, there exists $U_1^{(\ell,\ell')}, U_2^{(\ell,\ell')} \in \mathbb{R}^{n\times r}$ such that $U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top$ is the (ϵ_0, r)-approximation of $\exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top \right)$.

Then, we have

$$
\begin{aligned}
&\left\| \sum_{\ell=1}^m \sum_{\ell'=1}^m \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right)^{\circ C^{(T_\ell,T_{\ell'})}} - \sum_{\ell=1}^m \sum_{\ell'=1}^m \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right)^{\circ C^{(T_\ell,T_{\ell'})}} \right\|_1 \\
\leq\ & O\left(\frac{n^2 \epsilon_0 e^{o(B^2)}}{\log\left(n\right) \log\left(\epsilon+1\right)} \right).
\end{aligned}
$$

Proof. Now, we consider the ℓ_1 error. We have

$$
\begin{aligned}
&\left\| \sum_{\ell=1}^m \sum_{\ell'=1}^m \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right)^{\circ C^{(T_\ell,T_{\ell'})}} - \sum_{\ell=1}^m \sum_{\ell'=1}^m \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right)^{\circ C^{(T_\ell,T_{\ell'})}} \right\|_1 \\
=\ & \sum_{i=1}^n \sum_{j=1}^n \left(\sum_{\ell=1}^m \sum_{\ell'=1}^m \left(\exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right)^{\circ C^{(T_\ell,T_{\ell'})}} - \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right)^{\circ C^{(T_\ell,T_{\ell'})}} \right) \right)_{i,j}
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\left(\exp\left(Q^{(\ell)}\left(K^{(\ell')}\right)^\top/d\right)_{i,j}^{C^{(T_\ell,T_{\ell'})}} - \left(U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top\right)_{i,j}^{C^{(T_\ell,T_{\ell'})}}\right)\\
&= \sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\exp\left(Q^{(\ell)}\left(K^{(\ell')}\right)^\top/d\right)_{i,j}^{C^{(T_\ell,T_{\ell'})}} - \left(U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top\right)_{i,j}^{C^{(T_\ell,T_{\ell'})}}\right)\\
&\le \sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\left(\left|\mathrm{supp}\left(\exp\left(Q^{(\ell)}\left(K^{(\ell')}\right)^\top/d\right)^{\circ C^{(T_\ell,T_{\ell'})}}\right)\right| C^{(T_\ell,T_{\ell'})}\beta^{C^{(T_\ell,T_{\ell'})}-1}\epsilon_0\right)\\
&= \epsilon_0\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\left(\left|\mathrm{supp}\left(A^{(\ell,\ell')}\right)\right|\frac{b^2(1+\epsilon)^{\ell+\ell'}}{\log n}n^{o\left(\frac{b^2(1+\epsilon)^{\ell+\ell'}}{\log n}\right)}\right),
\end{aligned}
$$

where the first step follows from definition of ℓ_1 norm, the third step follows from exchanging the order of the summation, the fourth step follows from the mean value theorem (see Fact G.4) and the fact that there are

$$
\left|\mathrm{supp}\left(\exp\left(Q^{(\ell)}\left(K^{(\ell')}\right)^\top/d\right)^{\circ C^{(T_\ell,T_{\ell'})}}\right)\right|
$$

amount of non-zero entries to approximate, and the last step follows from the definition of $A^{(\ell,\ell')}$, β, and $C^{(T_\ell,T_{\ell'})}$ (see from the lemma statement).

Furthermore, by using the Cauchy–Schwarz inequality (see Fact J.3), we can get:

$$
\begin{aligned}
&\left\|\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\exp\left(Q^{(\ell)}\left(K^{(\ell')}\right)^\top/d\right)^{\circ C^{(T_\ell,T_{\ell'})}} - \sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\left(U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top\right)^{\circ C^{(T_\ell,T_{\ell'})}}\right\|_1\\
&= \epsilon_0\left(\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\left|\mathrm{supp}\left(A^{(\ell,\ell')}\right)\right|^2\right)^{1/2}\cdot\left(\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\frac{b^4(1+\epsilon)^{2\ell+2\ell'}}{\log^2 n}n^{o\left(\frac{b^2(1+\epsilon)^{\ell+\ell'}}{\log n}\right)}\right)^{1/2}\\
&\le \epsilon_0\left(\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\left|\mathrm{supp}\left(A^{(\ell,\ell')}\right)\right|\right)\cdot\left(\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\frac{b^4(1+\epsilon)^{2\ell+2\ell'}}{\log^2 n}n^{o\left(\frac{b^2(1+\epsilon)^{\ell+\ell'}}{\log n}\right)}\right)^{1/2}\\
&\le \epsilon_0\left(\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\left|\mathrm{supp}\left(A^{(\ell,\ell')}\right)\right|\right)\cdot\left(\int_1^{m+1}\int_1^{m+1}\frac{b^4(1+\epsilon)^{2x+2y}}{\log^2 n}n^{o\left(\frac{b^2(1+\epsilon)^{x+y}}{\log n}\right)}\,\mathrm{d}x\,\mathrm{d}y\right)^{1/2},
\end{aligned}
\tag{16}
$$

where the second step follows from Fact J.3 and the third step follows from the definition of Riemann integral.

Since $\{A^{(\ell,\ell')}\}_{\ell,\ell'\in[m]}$ forms a support basis of $QK^\top$ (see Lemma F.4), by applying Fact J.2, we have

$$
\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\left|\mathrm{supp}\left(A^{(\ell,\ell')}\right)\right| \le n^2. \tag{17}
$$

Additionally, for all $c > 0$, we have

$$
\begin{aligned}
&\int_1^{m+1}\frac{b^4(1+\epsilon)^{2x+2y}}{\log^2 n}n^{o\left(\frac{b^2(1+\epsilon)^{x+y}}{\log n}\right)}\,\mathrm{d}x\\
&\le \frac{cb^2(1+\epsilon)^{y+m+1}e^{b^2c(\epsilon+1)^{y+m+1}} - cb^2(1+\epsilon)^{y+1}e^{b^2c(\epsilon+1)^{y+1}} - e^{b^2c(\epsilon+1)^{y+m+1}} + e^{b^2c(\epsilon+1)^{y+1}}}{c^2\log^2 n\log(\epsilon+1)}\\
&= O\left(\frac{b^2(1+\epsilon)^{y+m+1}e^{b^2c(\epsilon+1)^{y+m+1}}}{c\log^2(n)\log(\epsilon+1)}\right),
\end{aligned}
\tag{18}
$$

where the first step follows from **Part 1** of Fact J.4 and the second step follows from the fact that $cb^2 (1+\epsilon)^{y+m+1} e^{b^2 c(\epsilon+1)^{y+m+1}}$ is the dominating term in the numerator.

We can further get

$$\begin{aligned}
\int_1^{m+1} \int_1^{m+1} \frac{b^4 (1+\epsilon)^{2x+2y}}{\log^2 n} n^{o\left(\frac{b^2(1+\epsilon)^{x+y}}{\log n}\right)} \mathrm{d}x \, \mathrm{d}y \leq & \int_1^{m+1} O\left(\frac{b^2 (1+\epsilon)^{y+m+1} e^{b^2 c(\epsilon+1)^{y+m+1}}}{c \log^2(n) \log(\epsilon+1)} \right) \mathrm{d}y \\
= & O\left(\int_1^{m+1} \frac{b^2 (1+\epsilon)^{y+m+1} e^{b^2 c(\epsilon+1)^{y+m+1}}}{c \log^2(n) \log(\epsilon+1)} \mathrm{d}y \right) \\
= & O\left(\frac{e^{b^2 c(\epsilon+1)^{2m+2}} - e^{b^2 c(\epsilon+1)^{m+2}}}{c^2 \log^2(n) \log^2(\epsilon+1)} \right) \\
= & O\left(\frac{e^{B^2 c(\epsilon+1)^2}}{c^2 \log^2(n) \log^2(\epsilon+1)} \right), \qquad (19)
\end{aligned}$$

where the first step follows from Eq. (18), the third step follows from **Part 2** of Fact J.4, and the last step follows from the fact that $e^{b^2 c(\epsilon+1)^{2m+2}}$ is the dominating term in numerator.

Combining Eq. (16), Eq. (17), and Eq. (19) together, we have

$$\begin{aligned}
& \left\| \sum_{\ell=1}^m \sum_{\ell'=1}^m \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right)^{\circ C^{(T_\ell, T_{\ell'})}} - \sum_{\ell=1}^m \sum_{\ell'=1}^m \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right)^{\circ C^{(T_\ell, T_{\ell'})}} \right\|_1 \\
\leq & O\left(\frac{n^2 \epsilon_0 e^{o(B^2)}}{\log(n) \log(\epsilon+1)} \right)
\end{aligned}$$

□

Then, we analyze the case where we do not use the bucketing technique.

Lemma G.11. *Let n, m, d be positive integers. Let $\epsilon \in (0,1)$. Let $Q = \sum_{\ell=1}^m Q^{(T_\ell)} \in \mathbb{R}^{n \times d}$ and $K = \sum_{\ell'=1}^m K^{(T_{\ell'})} \in \mathbb{R}^{n \times d}$ be defined as in Definition G.1, with $b = \min_{i \in [n], j \in [d]} \{|Q_{i,j}|, |K_{i,j}|\}$, $B = \max_{i \in [n], j \in [d]} \{|Q_{i,j}|, |K_{i,j}|\}$, and for all $\ell \in [m] \cup \{0\}$, we let $T_\ell = b(1+\epsilon)^\ell$. Let $A^{(T_\ell, T_{\ell'})} = Q^{(T_\ell)} \left(K^{(T_{\ell'})}\right)^\top = C^{(T_\ell, T_{\ell'})} \cdot Q^{(\ell)} \left(K^{(\ell')}\right)^\top \in \mathbb{R}^{n \times n}$. Let $\beta = \max\{\|Q\|_\infty, \|K\|_\infty\}$.*

Suppose $m = 1$, which implies

$$\exp\left(QK^\top / d\right) = \exp\left(Q^{(m)} \left(K^{(m)}\right)^\top / d \right)^{\circ C^{(T_m, T_m)}},$$

where $C^{(T_m, T_m)} = B^2 / \log n$.

Suppose by Lemma A.4, there exists $U_1^{(m,m)}, U_2^{(m,m)} \in \mathbb{R}^{n \times r}$ such that $U_1^{(m,m)} \left(U_2^{(m,m)}\right)^\top$ is the (ϵ_0, r)-approximation of $\exp\left(Q^{(m)} \left(K^{(m)}\right)^\top / d \right)$.

Then, we have

$$\left\| \exp\left(QK^\top / d\right) - \left(U_1^{(m,m)} \left(U_2^{(m,m)}\right)^\top \right)^{\circ C^{(T_m, T_m)}} \right\|_1 \leq O\left(\frac{n^2 B^2}{\log n} e^{o(B^2)} \epsilon_0 \right).$$

Proof. We have

$$\left\| \exp\left(QK^\top / d\right) - \left(U_1^{(m,m)} \left(U_2^{(m,m)}\right)^\top \right)^{\circ C^{(T_m, T_m)}} \right\|_1$$

$$
\begin{aligned}
&= \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\exp\left(QK^\top/d\right)_{i,j} - \left(U_1^{(m,m)}\left(U_2^{(m,m)}\right)^\top\right)^{C^{(T_m,T_m)}}_{i,j}\right)\\
&\le \sum_{i=1}^{n}\sum_{j=1}^{n} O\left(\frac{B^2}{\log n}e^{o(B^2)}\epsilon_0\right)\\
&= O\left(\frac{n^2B^2}{\log n}e^{o(B^2)}\epsilon_0\right),
\end{aligned}
$$

where the first step follows from the definition of ℓ_1 norm and the second step follows from Lemma G.9. □

G.6 Bucketing Reduces the Error in ℓ_p Norm

Lemma G.12. *Let n, m, d be positive integers where $2 \le m \le n$. Let $\epsilon \in (0,1)$. Let $Q = \sum_{\ell=1}^{m} Q^{(T_\ell)} \in \mathbb{R}^{n\times d}$ and $K = \sum_{\ell'=1}^{m} K^{(T_{\ell'})} \in \mathbb{R}^{n\times d}$ be defined as in Definition G.1, with $b = \min_{i\in[n],j\in[d]}\{|Q_{i,j}|, |K_{i,j}|\}$, $B = \max_{i\in[n],j\in[d]}\{|Q_{i,j}|, |K_{i,j}|\}$, and for all $\ell \in [m]\cup\{0\}$, we let $T_\ell = b(1+\epsilon)^\ell$. Let $A^{(T_\ell,T_{\ell'})} = Q^{(T_\ell)}\left(K^{(T_{\ell'})}\right)^\top = C^{(T_\ell,T_{\ell'})}\cdot Q^{(\ell)}\left(K^{(\ell')}\right)^\top \in \mathbb{R}^{n\times n}$.*

Let $m := \lfloor \log_{1+\epsilon}(B/b)\rfloor + 1$ and $C^{(T_\ell,T_{\ell'})} = \frac{b^2(1+\epsilon)^{\ell+\ell'}}{\log n}$. Let $\beta = \max\{\|Q\|_\infty, \|K\|_\infty\}$.

Suppose by Lemma A.4, there exists $U_1^{(\ell,\ell')}, U_2^{(\ell,\ell')} \in \mathbb{R}^{n\times r}$ such that $U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top$ is the (ϵ_0, r)-approximation of $\exp\left(Q^{(\ell)}\left(K^{(\ell')}\right)^\top\right)$.

Then, we can get a tighter bound for

$$
\left\|\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\exp\left(Q^{(\ell)}\left(K^{(\ell')}\right)^\top/d\right)^{\circ C^{(T_\ell,T_{\ell'})}} - \sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\left(U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top\right)^{\circ C^{(T_\ell,T_{\ell'})}}\right\|_p
$$

compared with

$$
\left\|\exp\left(QK^\top/d\right) - \left(U_1^{(m,m)}\left(U_2^{(m,m)}\right)^\top\right)^{\circ C^{(T_m,T_m)}}\right\|_p.
$$

Proof. Now, we consider the ℓ_p error.

Using **Part 2** of Fact J.9, we have

$$
\begin{aligned}
&\left\|\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\exp\left(Q^{(\ell)}\left(K^{(\ell')}\right)^\top/d\right)^{\circ C^{(T_\ell,T_{\ell'})}} - \sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\left(U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top\right)^{\circ C^{(T_\ell,T_{\ell'})}}\right\|_p^p\\
&= \sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\left\|\exp\left(Q^{(\ell)}\left(K^{(\ell')}\right)^\top/d\right)^{\circ C^{(T_\ell,T_{\ell'})}} - \left(U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top\right)^{\circ C^{(T_\ell,T_{\ell'})}}\right\|_p^p. \qquad (20)
\end{aligned}
$$

Below, we apply the same multi-threshold support basis for $\exp\left(QK^\top/d\right)$, but for each matrix, we take out the constant $C^{(T_m,T_m)}$ to simulate the case where we do not use bucketing.

Therefore, we can get

$$
\left\|\exp\left(QK^\top/d\right) - \left(U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top\right)^{\circ C^{(T_m,T_m)}}\right\|_p^p
$$

$$
\begin{aligned}
&= \left\| \sum_{\ell=1}^m \sum_{\ell'=1}^m \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right)^{\circ C^{(T_m,T_m)}} - \sum_{\ell=1}^m \sum_{\ell'=1}^m \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right)^{\circ C^{(T_m,T_m)}} \right\|_p^p \\
&= \left\| \sum_{\ell=1}^m \sum_{\ell'=1}^m \left(\exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right)^{\circ C^{(T_m,T_m)}} - \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right)^{\circ C^{(T_m,T_m)}} \right) \right\|_p^p \\
&\leq \sum_{\ell=1}^m \sum_{\ell'=1}^m \left\| \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right)^{\circ C^{(T_m,T_m)}} - \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right)^{\circ C^{(T_m,T_m)}} \right\|_p^p,
\end{aligned}
$$

where the second step follows from the distributive law and the third step follows from **Part 2** of Fact J.9. Applying **Part 1** and **Part 3** of Fact J.9, we can get

$$
\left\| A^{\circ k} + B^{\circ k} \right\|_p^p = \left\| (A+B)^{\circ k} \right\|_p^p \leq \|A+B\|_p^{pk},
$$

and for each ℓ and ℓ', we can use it to show:

$$
\begin{aligned}
& \left\| \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right)^{\circ C^{(T_\ell,T_{\ell'})}} - \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right)^{\circ C^{(T_\ell,T_{\ell'})}} \right\|_p^p \\
\leq & \left\| \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right) - \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right) \right\|_p^{p \cdot C^{(T_\ell,T_{\ell'})}}
\end{aligned}
$$

which is tighter compared with

$$
\begin{aligned}
& \left\| \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right)^{\circ C^{(T_m,T_m)}} - \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right)^{\circ C^{(T_m,T_m)}} \right\|_p^p \\
\leq & \left\| \exp\left(Q^{(\ell)} \left(K^{(\ell')}\right)^\top / d \right) - \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')}\right)^\top \right) \right\|_p^{p \cdot C^{(T_m,T_m)}}.
\end{aligned}
$$

□

Then, we analyze the case where we do not use the bucketing technique.

Lemma G.13. *Let n, m, d be positive integers. Let $\epsilon \in (0,1)$. Let $Q = \sum_{\ell=1}^m Q^{(T_\ell)} \in \mathbb{R}^{n \times d}$ and $K = \sum_{\ell'=1}^m K^{(T_{\ell'})} \in \mathbb{R}^{n \times d}$ be defined as in Definition G.1, with $b = \min_{i \in [n], j \in [d]} \{|Q_{i,j}|, |K_{i,j}|\}$, $B = \max_{i \in [n], j \in [d]} \{|Q_{i,j}|, |K_{i,j}|\}$, and for all $\ell \in [m] \cup \{0\}$, we let $T_\ell = b(1+\epsilon)^\ell$. Let $A^{(T_\ell,T_{\ell'})} = Q^{(T_\ell)} \left(K^{(T_{\ell'})}\right)^\top = C^{(T_\ell,T_{\ell'})} \cdot Q^{(\ell)} \left(K^{(\ell')}\right)^\top \in \mathbb{R}^{n \times n}$. Let $\beta = \max\{\|Q\|_\infty, \|K\|_\infty\}$.*

Suppose $m = 1$, which implies

$$
\exp\left(QK^\top / d\right) = \exp\left(Q^{(m)} \left(K^{(m)}\right)^\top / d \right)^{\circ C^{(T_m,T_m)}},
$$

where $C^{(T_m,T_m)} = B^2 / \log n$.

Suppose by Lemma A.4, there exists $U_1^{(m,m)}, U_2^{(m,m)} \in \mathbb{R}^{n \times r}$ such that $U_1^{(m,m)} \left(U_2^{(m,m)}\right)^\top$ is the (ϵ_0, r)-approximation of $\exp\left(Q^{(m)} \left(K^{(m)}\right)^\top / d \right)$.

Then, we have

$$\left\| \exp\left(QK^\top/d\right) - \left(U_1^{(m,m)} \left(U_2^{(m,m)}\right)^\top\right)^{\circ C^{(T_m,T_m)}} \right\|_p \leq O\left(\frac{n^2 B^2}{\log n} e^{o(B^2)} \epsilon_0\right).$$

Proof. We have

$$\begin{aligned}
& \left\| \exp\left(QK^\top/d\right) - \left(U_1^{(m,m)} \left(U_2^{(m,m)}\right)^\top\right)^{\circ C^{(T_m,T_m)}} \right\|_p^p \\
\leq\ & |\mathrm{supp}\,(A)| \cdot \left\| \exp\left(QK^\top/d\right) - \left(U_1^{(m,m)} \left(U_2^{(m,m)}\right)^\top\right)^{\circ C^{(T_m,T_m)}} \right\|_\infty^p \\
\leq\ & |\mathrm{supp}\,(A)| \cdot O\left(\frac{B^2}{\log n} e^{o(B^2)} \epsilon_0\right)^p,
\end{aligned}$$

where the first step follows from the definition of ℓ_1 norm and the second step follows from Lemma G.9.

Therefore, we can get

$$\left\| \exp\left(QK^\top/d\right) - \left(U_1^{(m,m)} \left(U_2^{(m,m)}\right)^\top\right)^{\circ C^{(T_m,T_m)}} \right\|_p \leq O\left(\frac{n^{2/p} B^2}{\log n} e^{o(B^2)} \epsilon_0\right)$$

□

H SKETCHING REDUCES THE TIME COMPLEXITY OF POLYNOMIAL ATTENTION

We note that Kacham et al. (2024) only analyzes the Frobenius norm error of the sketching matrix. In this section, we find the ℓ_∞ error guarantee.

In Appendix H.1, we express the two-vector JL moment property from Ahle et al. (2020) into the probabilistic statement. In Appendix H.2, we use the sketching technique to show that for each i, j, the inner product of the sketched vectors $\left\langle \phi'\left((U_1)_{i,*}\right), \phi'\left((U_2)_{j,*}\right)\right\rangle$ is close to the original un-sketched polynomial kernel $\left\langle (U_1)_{i,*}, (U_2)_{j,*}\right\rangle^p$. In Appendix H.3, we use the union bound over all i and j to show that the ℓ_∞ norm of $\left(U_1 U_2^\top\right)^{\circ p} - \phi'(U_1)\phi'(U_2)^\top$ is upper bounded, where the randomized feature mapping ϕ' is applied row-wisely to matrices U_1 and U_2.

H.1 Probabilistic Statement of Two Vector JL Moment Property

We formalize the guarantee provided by random projections. This guarantees approximately preserving the inner products between vectors with high probability. This result is a direct consequence of the JL moment property (Definition A.11) and plays a crucial role in ensuring that geometric relationships between vectors are maintained under dimensionality reduction:

Lemma H.1 (Inner Product Preservation)**.** *Let $S \in \mathbb{R}^{m \times d}$ be a random matrix satisfying the (ϵ, δ, t)-JL moment property for some $\epsilon, \delta > 0$ and integer $t \geq 1$. Then for all vectors $x, y \in \mathbb{R}^d$, with probability at least $1 - \delta$, we have:*

$$\left|(Sx)^\top (Sy) - x^\top y\right| \leq \epsilon \|x\|_2 \|y\|_2.$$

Proof. By Lemma A.12, we can get

$$\left\|(Sx)^\top (Sy) - x^\top y\right\|_{L_t} \leq \epsilon \delta^{1/t} \|x\|_2 \|y\|_2. \tag{21}$$

By Definition A.11, we can get

$$\left\|(Sx)^\top (Sy) - x^\top y\right\|_{L_t} = \mathbb{E}\left[\left|(Sx)^\top (Sy) - x^\top y\right|^t\right]^{1/t}. \tag{22}$$

Combining Eq. (21) and Eq. (22), we can get

$$\mathbb{E}\left[\left|(Sx)^\top(Sy) - x^\top y\right|^t\right] \leq \epsilon^t \delta \|x\|_2^t \|y\|_2^t.$$

Using Markov's inequality (see Fact J.7), we have

$$\begin{aligned}
\Pr\left[\left|(Sx)^\top(Sy) - x^\top y\right| > \epsilon\|x\|_2\|y\|_2\right] = & \Pr\left[\left|(Sx)^\top(Sy) - x^\top y\right|^t > \epsilon^t\|x\|_2^t\|y\|_2^t\right] \\
\leq & \frac{\mathbb{E}\left[\left|(Sx)^\top(Sy) - x^\top y\right|^t\right]}{\epsilon^t\|x\|_2^t\|y\|_2^t} \\
\leq & \delta.
\end{aligned}$$

□

H.2 Pointwise Approximation of Polynomial Kernel via Sketching

Now, we show that the randomized feature mapping ϕ' can approximate the value of a degree-p polynomial kernel between any pair of vectors up to a small additive error, with high probability:

Lemma H.2 (Pointwise Approximation of Polynomial Kernel via Sketching)**.** *Let $\{(U_1)_{i,*}\}_{i=1}^n, \{(U_2)_{j,*}\}_{j=1}^n \subset \mathbb{R}^r$. Let p be an even positive integer. Let $\epsilon \in (0,1)$ be the accuracy parameter and $\delta \in (0,1)$ be the failure probability.*

Then, there exists a randomized feature mapping

$$\phi' : \mathbb{R}^r \to \mathbb{R}^{z^2}, \quad \text{for } z = \Theta\left(p\epsilon^{-2}\log(1/\delta)\right)$$

defined as $\phi'(x) := \left(Sx^{\otimes(p/2)}\right)^{\otimes 2}$ where $S \in \mathbb{R}^{z \times r^{p/2}}$ is a sketching matrix, such that for all $i, j \in [n]$, the following hold with probability at least $1-\delta$:

$$\left|\left\langle \phi'\left((U_1)_{i,*}\right), \phi'\left((U_2)_{j,*}\right)\right\rangle - \left\langle (U_1)_{i,*}, (U_2)_{j,*}\right\rangle^p\right| \leq \epsilon \|\,(U_1)_{i,*}\,\|_2^p \|\,(U_2)_{j,*}\,\|_2^p.$$

Proof. We have

$$\begin{aligned}
& \left|\left\langle \phi'\left((U_1)_{i,*}\right), \phi'\left((U_2)_{j,*}\right)\right\rangle - \left\langle (U_1)_{i,*}, (U_2)_{j,*}\right\rangle^p\right| \\
= & \left|\left\langle \left(S\,(U_1)_{i,*}^{\otimes(p/2)}\right)^{\otimes 2}, \left(S\,(U_2)_{j,*}^{\otimes(p/2)}\right)^{\otimes 2}\right\rangle - \left\langle (U_1)_{i,*}, (U_2)_{j,*}\right\rangle^p\right| \\
= & \left|\left\langle S\,(U_1)_{i,*}^{\otimes(p/2)}, S\,(U_2)_{j,*}^{\otimes(p/2)}\right\rangle^2 - \left\langle (U_1)_{i,*}, (U_2)_{j,*}\right\rangle^p\right| \\
= & \left|\left(\left(S\,(U_1)_{i,*}^{\otimes(p/2)}\right)^\top \left(S\,(U_2)_{j,*}^{\otimes(p/2)}\right)\right)^2 - \left(\left\langle (U_1)_{i,*}, (U_2)_{j,*}\right\rangle^{p/2}\right)^2\right| \\
= & \left|\left(S\,(U_1)_{i,*}^{\otimes(p/2)}\right)^\top \left(S\,(U_2)_{j,*}^{\otimes(p/2)}\right) - \left\langle (U_1)_{i,*}, (U_2)_{j,*}\right\rangle^{p/2}\right| \\
& \cdot \left|\left(S\,(U_1)_{i,*}^{\otimes(p/2)}\right)^\top \left(S\,(U_2)_{j,*}^{\otimes(p/2)}\right) + \left\langle (U_1)_{i,*}, (U_2)_{j,*}\right\rangle^{p/2}\right|,
\end{aligned}$$

where the first step follows from the definition of ϕ' (see from the lemma statement), the second step follows from **Part 1** of Fact J.6, the third step follows from $\langle a, b\rangle = a^\top b$ (see Fact J.3), and the last step follows from $a^2 - b^2 = (a+b)(a-b)$.

Considering $\left|\left(S\,(U_1)_{i,*}^{\otimes(p/2)}\right)^\top \left(S\,(U_2)_{j,*}^{\otimes(p/2)}\right) - \left\langle (U_1)_{i,*}, (U_2)_{j,*}\right\rangle^{p/2}\right|$, with probability $1-\delta$, we have

$$\left|\left(S\,(U_1)_{i,*}^{\otimes(p/2)}\right)^\top \left(S\,(U_2)_{j,*}^{\otimes(p/2)}\right) - \left\langle (U_1)_{i,*}, (U_2)_{j,*}\right\rangle^{p/2}\right|$$

$$
\begin{aligned}
&= \left| \left(S \left(U_1 \right)_{i,*}^{\otimes (p/2)} \right)^\top \left(S \left(U_2 \right)_{j,*}^{\otimes (p/2)} \right) - \left\langle \left(U_1 \right)_{i,*}^{\otimes p/2}, \left(U_2 \right)_{j,*}^{\otimes p/2} \right\rangle \right| \\
&= \left| \left(S \left(U_1 \right)_{i,*}^{\otimes (p/2)} \right)^\top \left(S \left(U_2 \right)_{j,*}^{\otimes (p/2)} \right) - \left(\left(U_1 \right)_{i,*}^{\otimes p/2} \right)^\top \left(U_2 \right)_{j,*}^{\otimes p/2} \right| \\
&\leq \epsilon \left\| \left(U_1 \right)_{i,*}^{\otimes p/2} \right\|_2 \left\| \left(U_2 \right)_{j,*}^{\otimes p/2} \right\|_2 \\
&= \epsilon \left\| \left(U_1 \right)_{i,*} \right\|_2^{p/2} \left\| \left(U_2 \right)_{j,*} \right\|_2^{p/2},
\end{aligned}
$$

where the first step follows from **Part 2** of Fact J.6, the second step follows from Fact J.3, the third step follows from Lemma H.1, and the last step follows from **Part 3** of Fact J.6.

Now, we consider $\left| \left(S \left(U_1 \right)_{i,*}^{\otimes (p/2)} \right)^\top \left(S \left(U_2 \right)_{j,*}^{\otimes (p/2)} \right) + \left\langle \left(U_1 \right)_{i,*}, \left(U_2 \right)_{j,*} \right\rangle^{p/2} \right|$: we have with probability $1 - \delta$,

$$
\begin{aligned}
&\left| \left(S \left(U_1 \right)_{i,*}^{\otimes (p/2)} \right)^\top \left(S \left(U_2 \right)_{j,*}^{\otimes (p/2)} \right) + \left\langle \left(U_1 \right)_{i,*}, \left(U_2 \right)_{j,*} \right\rangle^{p/2} \right| \\
= &\left| \left(S \left(U_1 \right)_{i,*}^{\otimes (p/2)} \right)^\top \left(S \left(U_2 \right)_{j,*}^{\otimes (p/2)} \right) - \left\langle \left(U_1 \right)_{i,*}, \left(U_2 \right)_{j,*} \right\rangle^{p/2} + 2 \left\langle \left(U_1 \right)_{i,*}, \left(U_2 \right)_{j,*} \right\rangle^{p/2} \right| \\
\leq &\left| \left(S \left(U_1 \right)_{i,*}^{\otimes (p/2)} \right)^\top \left(S \left(U_2 \right)_{j,*}^{\otimes (p/2)} \right) - \left\langle \left(U_1 \right)_{i,*}, \left(U_2 \right)_{j,*} \right\rangle^{p/2} \right| + 2 \left| \left\langle \left(U_1 \right)_{i,*}, \left(U_2 \right)_{j,*} \right\rangle^{p/2} \right| \\
\leq &\epsilon \left\| \left(U_1 \right)_{i,*} \right\|_2^{p/2} \left\| \left(U_2 \right)_{j,*} \right\|_2^{p/2} + 2 \left| \left\langle \left(U_1 \right)_{i,*}, \left(U_2 \right)_{j,*} \right\rangle^{p/2} \right| \\
\leq &\epsilon \left\| \left(U_1 \right)_{i,*} \right\|_2^{p/2} \left\| \left(U_2 \right)_{j,*} \right\|_2^{p/2} + 2 \left\| \left(U_1 \right)_{i,*} \right\|_2^{p/2} \left\| \left(U_2 \right)_{j,*} \right\|_2^{p/2},
\end{aligned}
$$

where the second step follows from the triangle inequality (see Fact J.3), the third step follows from Lemma H.1, and the last step follows from the Cauchy-Schwarz inequality (see Fact J.3).

Thus, combining everything together, we have that

$$
\left| \left\langle \phi' \left(\left(U_1 \right)_{i,*} \right), \phi' \left(\left(U_2 \right)_{j,*} \right) \right\rangle - \left\langle \left(U_1 \right)_{i,*}, \left(U_2 \right)_{j,*} \right\rangle^p \right| \leq \epsilon \| \left(U_1 \right)_{i,*} \|_2^p \| \left(U_2 \right)_{j,*} \|_2^p
$$

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

□

H.3 Union Bound to All Entries

We generalize the previous pointwise sketching result (Lemma H.2) by providing an ℓ_∞ bound over the entire polynomial kernel matrix. It shows that, with high probability, the randomized sketching method can approximate the full matrix $\left(U_1 U_2^\top \right)^{\circ p}$ entry-wise up to a controllable additive error, uniformly across all pairs of rows:

Lemma H.3 (ℓ_∞ Guarantee for Polynomial Sketching)**.** *Let $\{(U_1)_{i,*}\}_{i=1}^n, \{(U_2)_{j,*}\}_{j=1}^n \subset \mathbb{R}^r$. Fix an even integer $p \geq 2$. Let $\epsilon \in (0,1)$ be the accuracy parameter and $\delta \in (0,1)$ be the failure probability. Let $\phi' : \mathbb{R}^r \to \mathbb{R}^{z^2}$ be a randomized feature mapping constructed using the sketching technique in Theorem A.9, with sketch size $z = \Theta(p\epsilon^{-2} \log(n/\delta))$ and $\phi' \left((U_1)_{i,*} \right) := \left(S \left((U_1)_{i,*} \right)^{\otimes (p/2)} \right)^{\otimes 2}$. For all matrix $A \in \mathbb{R}^{n \times r}$, we write $\phi'(A) \in \mathbb{R}^{n \times z^2}$ to denote ϕ' being applied to A row-wisely.*

Then, with probability at least $1 - \delta$, we have

$$
\left\| \left(U_1 U_2^\top \right)^{\circ p} - \phi'(U_1) \phi'(U_2)^\top \right\|_\infty \leq \max_{i,j \in [n]} \left\{ \epsilon \left\| \left(U_1 \right)_{i,*} \right\|_2^p \left\| \left(U_2 \right)_{j,*} \right\|_2^p \right\}.
$$

Proof. By Lemma H.2, we know that for each $i, j \in [n]$, the randomized mapping ϕ' satisfies

$$
\Pr \left[\left| \left\langle \phi' \left(\left(U_1 \right)_{i,*} \right), \phi' \left(\left(U_2 \right)_{j,*} \right) \right\rangle - \left\langle \left(U_1 \right)_{i,*}, \left(U_2 \right)_{j,*} \right\rangle^p \right| > \epsilon \left\| \left(U_1 \right)_{i,*} \right\|_2^p \cdot \left\| \left(U_2 \right)_{j,*} \right\|_2^p \right] \leq \delta',
$$

for some small $\delta' \in (0,1)$.

Since $i,j \in [n]$, we have n^2 total pairs (i,j). Applying the union bound (see Fact J.8) over all $i,j \in [n]$, we have

$$\begin{aligned} & \Pr\left[\exists i,j \in [n], \left|\left\langle \phi'\left((U_1)_{i,*}\right), \phi'\left((U_2)_{j,*}\right)\right\rangle - \left\langle (U_1)_{i,*}, (U_2)_{j,*}\right\rangle^p\right| > \epsilon \left\|(U_1)_{i,*}\right\|_2^p \cdot \left\|(U_2)_{j,*}\right\|_2^p\right] \\ \leq & n^2 \cdot \delta'. \end{aligned}$$

Setting $\delta' = \delta/n^2$, we can get that, with probability at least $1-\delta$,

$$\max_{i,j\in[n]} \left|\left\langle \phi'\left((U_1)_{i,*}\right), \phi'\left((U_2)_{j,*}\right)\right\rangle - \left\langle (U_1)_{i,*}, (U_2)_{j,*}\right\rangle^p\right| \leq \max_{i,j\in[n]} \left\{\epsilon \left\|(U_1)_{i,*}\right\|_2^p \cdot \left\|(U_2)_{j,*}\right\|_2^p\right\}. \tag{23}$$

On the other hand, we have

$$\begin{aligned} & \max_{i,j\in[n]} \left|\left\langle \phi'\left((U_1)_{i,*}\right), \phi'\left((U_2)_{j,*}\right)\right\rangle - \left\langle (U_1)_{i,*}, (U_2)_{j,*}\right\rangle^p\right| \\ = & \max_{i,j\in[n]} \left|\left\langle \phi'\left(U_1\right)_{i,*}, \phi'\left(U_2\right)_{j,*}\right\rangle - \left\langle (U_1)_{i,*}, (U_2)_{j,*}\right\rangle^p\right| \\ = & \max_{i,j\in[n]} \left|\left(\phi'\left(U_1\right)\phi'\left(U_2\right)^\top\right)_{i,j} - \left(\left(U_1\right)\left(U_2^\top\right)\right)_{i,j}^p\right| \\ = & \max_{i,j\in[n]} \left|\phi'\left(U_1\right)\phi'\left(U_2\right)^\top - \left(\left(U_1\right)\left(U_2^\top\right)\right)^{\circ p}\right|_{i,j} \\ = & \left\|\left(U_1U_2^\top\right)^{\circ p} - \phi'(U_1)\phi'(U_2)^\top\right\|_\infty, \end{aligned} \tag{24}$$

where the first step follows from the fact that ϕ' is applied row-wisely, the second step follows from the definition of matrix multiplication, and the last step follows from the definition of ℓ_∞ norm.

Finally, we consider the sketch size. By Lemma H.2, to get pointwise approximation to the polynomial kernel, we need

$$\Theta\left(p\epsilon^{-2}\log(1/\delta')\right).$$

Now, since we use union bound and setting $\delta' = \delta/n^2$, we can get the sketch size

$$\begin{aligned} z &= \Theta\left(p\epsilon^{-2}\log(1/\left(\delta/n^2\right))\right) \\ &= \Theta\left(p\epsilon^{-2}\log(n/\delta)\right). \end{aligned} \tag{25}$$

Combining Eq. (23), (24), and (25), we finish the proof. □

I ATTENTION OPTIMIZATION VIA MULTI-THRESHOLD SUPPORT BASIS AND SKETCHING

Now, we combine everything together and present our main result using multi-threshold support basis.

In Appendix I.1, we give a basic definition. In Appendix I.2, we approximate the (Q,K)-softmax-attention matrix using multi-threshold support basis decomposition and sketching. In Appendix I.3, we analyze the running time for constructing $\phi'\left(U_1^{(\ell,\ell')}\right)$ and $\phi'\left(U_2^{(\ell,\ell')}\right)$. In Appendix I.4, we solve the batch Gaussian kernel density estimation problem (see Definition B.10) using multi-threshold support basis decomposition. Finally, in Appendix I.5, we present our main theorem, which states the running time and correctness of approximating the attention computation problem (see Definition 1.2) using multi-threshold support basis decomposition and sketching.

I.1 A Basic Definition

In this section, we provide basic definitions and clarify the meaning of all notation. We note that in previous sections, some notational abuse may have occurred—for example, the symbol ϵ may have been used both as the accuracy parameter for sketching and as the bucketing parameter. Our goal here is to unify all theoretical results to derive the main result. Therefore, we redefine the relevant notations to improve clarity.

Definition I.1. *Given the query and the key matrices $Q, K \in \mathbb{R}^{n\times d}$, we define the (Q,K)-softmax-attention matrix as $A = \exp(QK^\top/d)$. Let $Q = \sum_{\ell=1}^m Q^{(T_\ell)} \in \mathbb{R}^{n\times d}$ and $K = \sum_{\ell'=1}^m K^{(T_{\ell'})} \in \mathbb{R}^{n\times d}$ be defined as in Definition G.1, with $b = \min_{i\in[n],j\in[d]}\{|Q_{i,j}|, |K_{i,j}|\}$, $B = \max_{i\in[n],j\in[d]}\{|Q_{i,j}|, |K_{i,j}|\}$, and for all $\ell \in [m]\cup\{0\}$, we let $T_\ell = b(1+\epsilon_B)^\ell$, where ϵ_B is the bucketing parameter. Let $A^{(T_\ell,T_{\ell'})} = Q^{(T_\ell)}\left(K^{(T_{\ell'})}\right)^\top = C^{(T_\ell,T_{\ell'})} \cdot Q^{(\ell)}\left(K^{(\ell')}\right)^\top \in \mathbb{R}^{n\times n}$. Suppose by Lemma A.4, there exists $U_1^{(\ell,\ell')}, U_2^{(\ell,\ell')} \in \mathbb{R}^{n\times r}$ such that $U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top$ is the (ϵ_0, r)-approximation of $\exp\left(Q^{(\ell)}\left(K^{(\ell')}\right)^\top\right)$. Let $\phi' : \mathbb{R}^d \to \mathbb{R}^{r^2}$ be defined as $\phi'(q_i) := \left(Sq_i^{\otimes(p/2)}\right)^{\otimes 2}$, where $S \in \mathbb{R}^{r\times d^{p/2}}$ with $r = \Theta\left(p\epsilon^{-2}\log(1/\delta)\right)$ is a sketching matrix. With $\epsilon, \epsilon_0 \in (0,1)$, we let*

$$\epsilon_2 := O\left(\frac{B^2}{\log n} e^{o(B^2)}\epsilon_0 + \sum_{\ell=1}^m\sum_{\ell'=1}^m \max_{i,j\in[n]}\left\{\epsilon \left\|(U_1)_{i,*}\right\|_2^{C^{(T_\ell,T_{\ell'})}} \left\|(U_2)_{j,*}\right\|_2^{C^{(T_\ell,T_{\ell'})}}\right\}\right).$$

I.2 (Q,K)-Softmax-Attention Matrix Approximation

The (Q,K)-softmax-attention matrix $\exp(QK^\top/d)$ is central to transformer models, but its computation is costly due to the exponential operation and quadratic complexity. To reduce computational overhead, we approximate this matrix using a combination of polynomial kernel sketching and bucketing. The following lemma shows that by decomposing the input into multiple components and applying randomized feature mappings to each, we can uniformly approximate the entire softmax-attention matrix.

Lemma I.2 ((Q,K)-Softmax-Attention Matrix Approximation)**.** *Let everything be defined as in Definition I.1. Then, we can get*

$$\left\|\exp(QK^\top/d) - \sum_{\ell=1}^m\sum_{\ell'=1}^m \phi'\left(U_1^{(\ell,\ell')}\right)\phi'\left(U_2^{(\ell,\ell')}\right)^\top\right\|_\infty \le \epsilon_2.$$

Proof. We can get

$$\begin{aligned}
&\left\|\exp(QK^\top/d) - \sum_{\ell=1}^m\sum_{\ell'=1}^m \phi'\left(U_1^{(\ell,\ell')}\right)\phi'\left(U_2^{(\ell,\ell')}\right)^\top\right\|_\infty \\
=\ &\left\|\exp(QK^\top/d) - \sum_{\ell=1}^m\sum_{\ell'=1}^m \left(U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top\right)^{\circ C^{(T_\ell,T_{\ell'})}}\right. \\
&\left. + \sum_{\ell=1}^m\sum_{\ell'=1}^m \left(U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top\right)^{\circ C^{(T_\ell,T_{\ell'})}} - \sum_{\ell=1}^m\sum_{\ell'=1}^m \phi'\left(U_1^{(\ell,\ell')}\right)\phi'\left(U_2^{(\ell,\ell')}\right)^\top\right\|_\infty \\
\le\ &\left\|\exp(QK^\top/d) - \sum_{\ell=1}^m\sum_{\ell'=1}^m \left(U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top\right)^{\circ C^{(T_\ell,T_{\ell'})}}\right\|_\infty \\
&+ \left\|\sum_{\ell=1}^m\sum_{\ell'=1}^m \left(U_1^{(\ell,\ell')}\left(U_2^{(\ell,\ell')}\right)^\top\right)^{\circ C^{(T_\ell,T_{\ell'})}} - \sum_{\ell=1}^m\sum_{\ell'=1}^m \phi'\left(U_1^{(\ell,\ell')}\right)\phi'\left(U_2^{(\ell,\ell')}\right)^\top\right\|_\infty
\end{aligned}$$

$$\leq \left\| \exp(QK^\top/d) - \sum_{\ell=1}^m \sum_{\ell'=1}^m \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')} \right)^\top \right)^{\circ C^{(T_\ell, T_{\ell'})}} \right\|_\infty$$
$$+ \sum_{\ell=1}^m \sum_{\ell'=1}^m \left\| \left(U_1^{(\ell,\ell')} \left(U_2^{(\ell,\ell')} \right)^\top \right)^{\circ C^{(T_\ell, T_{\ell'})}} - \phi'\left(U_1^{(\ell,\ell')} \right) \phi'\left(U_2^{(\ell,\ell')} \right)^\top \right\|_\infty$$
$$\leq O\left(\frac{B^2}{\log n} e^{o(B^2)} \epsilon_0 + \sum_{\ell=1}^m \sum_{\ell'=1}^m \max_{i,j\in[n]} \left\{ \epsilon \left\| (U_1)_{i,*} \right\|_2^{C^{(T_\ell, T_{\ell'})}} \left\| (U_2)_{j,*} \right\|_2^{C^{(T_\ell, T_{\ell'})}} \right\} \right),$$

where the second and the third step follows from the triangle inequality (see Fact J.3), and the last step follows from combining Lemma G.8 and Lemma H.3. □

I.3 Running Time Analysis of Constructing $\phi'\left(U_1^{(\ell,\ell')}\right)$ and $\phi'\left(U_2^{(\ell,\ell')}\right)$

We analyze the runtime complexity of constructing the sketched feature representations used in approximating the softmax-attention matrix. The following lemma shows that the total time required to compute all randomized feature mappings $\phi'\left(U_1^{(\ell,\ell')}\right)$ and $\phi'\left(U_2^{(\ell,\ell')}\right)$, across all bucketed matrix pairs $(\ell,\ell') \in [m] \times [m]$, is almost linear in n:

Lemma I.3. *Let everything be defined as in Definition I.1.*

Then, it takes

$$O\left(\frac{B^6 \log^2(n/\delta)}{\epsilon^4} n^{1+o(1)} \right)$$

time to construct $\phi'\left(U_1^{(\ell,\ell')}\right)$ and $\phi'\left(U_2^{(\ell,\ell')}\right)$, for all $(\ell,\ell') \in [m] \times [m]$.

Proof. By Lemma A.2, for each $(\ell,\ell') \in [m] \times [m]$, it takes $O\left(n^{1+o(1)}\right)$ time to construct $U_1^{(\ell,\ell')} \in \mathbb{R}^{n\times r}$ and $U_2^{(\ell,\ell')} \in \mathbb{R}^{n\times r}$, with $r < n^{o(1)}$.

By Theorem A.9, with

$$z = \Theta\left(C^{(T_\ell, T_{\ell'})} \epsilon^{-2} \log(n/\delta) \right) \tag{26}$$

being the sketch size (see Lemma H.3), where $\epsilon \in (0, 0.5)$ and $\delta \in (0,1)$ respectively are the accuracy parameter and failure probability for sketching, since ϕ' is applied row-wisely to a matrix, computing $\phi'\left(U_1^{(\ell,\ell')}\right)$ and $\phi'\left(U_2^{(\ell,\ell')}\right)$ respectively requires n times of:

1. $\frac{C^{(T_\ell, T_{\ell'})}}{2}$ matrix-vector multiplications with matrices of size $r \times z$,
2. $\frac{C^{(T_\ell, T_{\ell'})}}{2} - 2$ matrix-vector multiplications with matrices of size $z \times z$,
3. $\frac{C^{(T_\ell, T_{\ell'})}}{2} - 1$ Hadamard products of z-dimensional vectors,
4. and 1 self-Kronecker product of an z-dimensional vector.

Step 1 takes

$$\frac{C^{(T_\ell, T_{\ell'})}}{2} \cdot rz = O\left(C^{(T_\ell, T_{\ell'})} rz \right).$$

Step 2 takes

$$\left(\frac{C^{(T_\ell,T_{\ell'})}}{2}-2\right)\cdot z^2 = O\left(C^{(T_\ell,T_{\ell'})}z^2\right).$$

Step 3 takes

$$\left(\frac{C^{(T_\ell,T_{\ell'})}}{2}-1\right)\cdot z = O\left(C^{(T_\ell,T_{\ell'})}z\right).$$

Step 4 takes

$$O\left(z^2\right).$$

Therefore, for each $(\ell,\ell')\in[m]\times[m]$, it takes

$$\begin{aligned}&O\left(n^{1+o(1)}+n\cdot\left(C^{(T_\ell,T_{\ell'})}rz+C^{(T_\ell,T_{\ell'})}z^2+C^{(T_\ell,T_{\ell'})}z+z^2\right)\right)\\ =\ &O\left(n^{1+o(1)}+nC^{(T_\ell,T_{\ell'})}\cdot\left(rz+z^2\right)\right).\end{aligned}$$

In total, it takes

$$\begin{aligned}&\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}O\left(n^{1+o(1)}+nC^{(T_\ell,T_{\ell'})}\cdot\left(rz+z^2\right)\right)\\ =\ &O\left(\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}n^{1+o(1)}+\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}nC^{(T_\ell,T_{\ell'})}\cdot\left(rz+z^2\right)\right)\\ =\ &O\left(m^2n^{1+o(1)}+nr\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}C^{(T_\ell,T_{\ell'})}z+n\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}C^{(T_\ell,T_{\ell'})}z^2\right)\end{aligned}\tag{27}$$

time to compute $\phi'\left(U_1^{(\ell,\ell')}\right),\phi'\left(U_2^{(\ell,\ell')}\right)\in\mathbb{R}^{n\times z^2}$ for all $(\ell,\ell')\in[m]\times[m]$.

Now, we analyze the first term of Eq. (27): $m^2n^{1+o(1)}$.

By Fact G.2, we have

$$m=\left\lfloor\log_{1+\epsilon_B}\left(B/b\right)\right\rfloor+1.\tag{28}$$

Therefore, by Eq. (28), we have

$$\begin{aligned}m^2n^{1+o(1)} =\ &\left(\left\lfloor\log_{1+\epsilon_B}\left(B/b\right)\right\rfloor+1\right)^2n^{1+o(1)}\\ =\ &O\left(\log^2_{1+\epsilon_B}\left(B/b\right)n^{1+o(1)}\right).\end{aligned}\tag{29}$$

Now, we analyze the second term of Eq. (27): $nr\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}C^{(T_\ell,T_{\ell'})}z$.

We have

$$\begin{aligned}nr\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}C^{(T_\ell,T_{\ell'})}\cdot z &= n^{1+o(1)}\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}C^{(T_\ell,T_{\ell'})}\cdot z\\ &= n^{1+o(1)}\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}C^{(T_\ell,T_{\ell'})}\cdot C^{(T_\ell,T_{\ell'})}\epsilon^{-2}\log(n/\delta)\\ &= n^{1+o(1)}\epsilon^{-2}\log(n/\delta)\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\left(C^{(T_\ell,T_{\ell'})}\right)^2\end{aligned}$$

$$
\begin{aligned}
&= n^{1+o(1)} \epsilon^{-2} \log(n/\delta) \sum_{\ell=1}^m \sum_{\ell'=1}^m \left(\frac{b^2 (1+\epsilon_B)^{\ell+\ell'}}{\log n} \right)^2 \\
&= \frac{b^4 n^{1+o(1)} \epsilon^{-2} \log(n/\delta)}{\log^2 n} \sum_{\ell=1}^m \sum_{\ell'=1}^m (1+\epsilon_B)^{2\ell+2\ell'} \\
&= \frac{b^4 n^{1+o(1)} \epsilon^{-2} \log(n/\delta)}{\log^2 n} \left(\sum_{\ell=1}^m (1+\epsilon_B)^{2\ell} \right)^2, \qquad (30)
\end{aligned}
$$

where the first step follows from $r = n^{o(1)}$ (see Lemma A.4), the second step follows from Eq. (26), the fourth step follows from Lemma G.3 and $\epsilon_B > 0$ denote the bucketing parameter, and the last step follows from $(1+\epsilon_B)^{2\ell+2\ell'} = (1+\epsilon_B)^{2\ell} \cdot (1+\epsilon_B)^{2\ell'}$.

Furthermore, we can see that

$$
\begin{aligned}
\sum_{\ell=1}^m (1+\epsilon_B)^{2\ell} &= \frac{(1+\epsilon_B)^2 \left(1 - (1+\epsilon_B)^{2m}\right)}{1 - (1+\epsilon_B)^2} \\
&= O\left((1+\epsilon_B)^{2m} \right), \qquad (31)
\end{aligned}
$$

where the first step follows from the geometric series formula.

Combining Eq. (30) and (31), we have

$$
\begin{aligned}
nr \sum_{\ell=1}^m \sum_{\ell'=1}^m C^{(T_\ell, T_{\ell'})} \cdot z &= O\left(\frac{b^4 \log(n/\delta) e^{4m\epsilon_B}}{\epsilon^2 \log^2 n} n^{1+o(1)} \right) \\
&= O\left(\frac{B^4 \log(n/\delta)}{\epsilon^2 \log^2 n} n^{1+o(1)} \right), \qquad (32)
\end{aligned}
$$

where the second step follows from Eq. (28).

Now, we analyze the third term of Eq. (27): $n \sum_{\ell=1}^m \sum_{\ell'=1}^m C^{(T_\ell, T_{\ell'})} z^2$.

We have

$$
\begin{aligned}
n \sum_{\ell=1}^m \sum_{\ell'=1}^m C^{(T_\ell, T_{\ell'})} \cdot z^2 &= n \sum_{\ell=1}^m \sum_{\ell'=1}^m C^{(T_\ell, T_{\ell'})} \cdot \left(C^{(T_\ell, T_{\ell'})} \epsilon^{-2} \log(n/\delta) \right)^2 \\
&= n \epsilon^{-4} \log^2(n/\delta) \sum_{\ell=1}^m \sum_{\ell'=1}^m \left(C^{(T_\ell, T_{\ell'})} \right)^3 \\
&= n \epsilon^{-4} \log^2(n/\delta) \sum_{\ell=1}^m \sum_{\ell'=1}^m \left(\frac{b^2 (1+\epsilon_B)^{\ell+\ell'}}{\log n} \right)^3 \\
&= \frac{b^6 n \epsilon^{-4} \log^2(n/\delta)}{\log^3 n} \sum_{\ell=1}^m \sum_{\ell'=1}^m (1+\epsilon_B)^{3\ell+3\ell'} \\
&= \frac{b^6 n \epsilon^{-4} \log^2(n/\delta)}{\log^3 n} \left(\sum_{\ell=1}^m (1+\epsilon_B)^{3\ell} \right)^2, \qquad (33)
\end{aligned}
$$

where the first step follows from Eq. (26), the third step follows from Lemma G.3 and $\epsilon_B > 0$ denote the bucketing parameter, and the last step follows from $(1+\epsilon_B)^{3\ell+3\ell'} = (1+\epsilon_B)^{3\ell} \cdot (1+\epsilon_B)^{3\ell'}$.

Furthermore, we can see that

$$
\sum_{\ell=1}^m (1+\epsilon_B)^{3\ell} = \frac{(1+\epsilon_B)^3 \left(1 - (1+\epsilon_B)^{3m}\right)}{1 - (1+\epsilon_B)^3}
$$

$$= O\left((1+\epsilon_B)^{3m}\right) \tag{34}$$

where the first step follows from the geometric series formula.

Combining Eq. (33) and (34), we have

$$\begin{aligned} n\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m} C^{(T_\ell,T_{\ell'})}\cdot z^2 &= O\left(\frac{b^6\log^2(n/\delta)e^{6m\epsilon_B}}{\epsilon^4\log^3 n}n\right) \\ &= O\left(\frac{B^6\log^2(n/\delta)}{\epsilon^4\log^3 n}n\right), \end{aligned} \tag{35}$$

where the second step follows from Eq. (28).

Finally, combining Eq. (27), (29), (32), and (35), we can get that it takes

$$O\left(\frac{B^6\log^2(n/\delta)}{\epsilon^4}n^{1+o(1)}\right)$$

time to compute $\phi'\left(U_1^{(\ell,\ell')}\right), \phi'\left(U_2^{(\ell,\ell')}\right) \in \mathbb{R}^{n\times z^2}$ for all $(\ell,\ell') \in [m]\times[m]$. □

I.4 (Batch) Gaussian Kernel Density Estimation

The following lemma shows that, by using the polynomial sketching and bucketing approach described earlier, we can approximate the batch Gaussian kernel density computation with provably small ℓ_∞ error. Moreover, the algorithm runs in almost linear time with respect to n:

Lemma I.4. *Let everything be defined as in Definition I.1. Then, with probability $1-\delta$, we can solve the (Batch) Gaussian kernel density estimation (Definition B.10) by outputting $\mathsf{S} \in \mathbb{R}^{n\times d}$ satisfying*

$$\|\mathsf{S} - AV\|_\infty < n\epsilon_2 \|V\|_\infty$$

in $O\left(\frac{B^6\log^2(n/\delta)}{\epsilon^4}n^{1+o(1)}\right)$ time.

Proof. **Proof of correctness.**

We can get

$$AV = \exp\left(QK^\top/d\right)V.$$

We define $\mathsf{S} := \left(\sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\phi'\left(U_1^{(\ell,\ell')}\right)\phi'\left(U_2^{(\ell,\ell')}\right)^\top\right)V$.

Therefore, we can get

$$\begin{aligned} \|AV-\mathsf{S}\|_\infty &= \left\|\exp(QK^\top/d)V - \sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\phi'\left(U_1^{(\ell,\ell')}\right)\phi'\left(U_2^{(\ell,\ell')}\right)^\top V\right\|_\infty \\ &= \left\|\left(\exp(QK^\top/d) - \sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\phi'\left(U_1^{(\ell,\ell')}\right)\phi'\left(U_2^{(\ell,\ell')}\right)^\top\right)V\right\|_\infty \\ &\le n\left\|\exp(QK^\top/d) - \sum_{\ell=1}^{m}\sum_{\ell'=1}^{m}\phi'\left(U_1^{(\ell,\ell')}\right)\phi'\left(U_2^{(\ell,\ell')}\right)^\top\right\|_\infty \|V\|_\infty \\ &\le n\epsilon_2\|V\|_\infty, \end{aligned}$$

where the second step follows from the distributive law, the third step follows from the definition of the ℓ_∞ norm, and the fourth step follows from Lemma I.2.

Proof of the running time.

By Lemma I.3, we note that to obtain $\phi'\left(U_1^{(\ell,\ell')}\right), \phi'\left(U_2^{(\ell,\ell')}\right) \in \mathbb{R}^{n \times z^2}$, it takes

$$O\left(\frac{B^6 \log^2(n/\delta)}{\epsilon^4} n^{1+o(1)}\right) \tag{36}$$

time.

Then, we compute

$$\underbrace{\phi'\left(U_2^{(\ell,\ell')}\right)^\top}_{z^2 \times n} \underbrace{V}_{n \times d},$$

for all $(\ell, \ell') \in [m] \times [m]$, which takes

$$\begin{aligned}
\sum_{\ell=1}^m \sum_{\ell'=1}^m O\left(z^2 nd\right) = & \sum_{\ell=1}^m \sum_{\ell'=1}^m O\left(\left(C^{(T_\ell, T_{\ell'})}\right)^2 \epsilon^{-4} \log^2(n/\delta) nd\right) \\
= & O\left(\epsilon^{-4} \log^2(n/\delta) n^{1+o(1)} \sum_{\ell=1}^m \sum_{\ell'=1}^m \left(C^{(T_\ell, T_{\ell'})}\right)^2\right) \\
= & O\left(\epsilon^{-4} \log^2(n/\delta) n^{1+o(1)} \sum_{\ell=1}^m \sum_{\ell'=1}^m \left(\frac{b^2 (1+\epsilon_B)^{\ell+\ell'}}{\log n}\right)^2\right) \\
= & O\left(\frac{b^4 \log^2(n/\delta)}{\epsilon^4} n^{1+o(1)} \sum_{\ell=1}^m \sum_{\ell'=1}^m (1+\epsilon_B)^{2\ell+2\ell'}\right) \\
= & O\left(\frac{b^4 \log^2(n/\delta)}{\epsilon^4} n^{1+o(1)} \left(\sum_{\ell=1}^m (1+\epsilon_B)^{2\ell}\right)^2\right) \\
= & O\left(\frac{b^4 \log^2(n/\delta) e^{4m\epsilon_B}}{\epsilon^4} n^{1+o(1)}\right) \\
= & O\left(\frac{B^4 \log^2(n/\delta)}{\epsilon^4} n^{1+o(1)}\right),
\end{aligned} \tag{37}$$

where the first step follows from Eq. (26), the second step follows from $d = O(\log n)$, the third step follows from Lemma G.3, the fifth step follows from $(1+\epsilon_B)^{2\ell+2\ell'} = (1+\epsilon_B)^{2\ell} \cdot (1+\epsilon_B)^{2\ell'}$, the sixth step follows from Eq. (31), and the last step follows from Eq. (28).

Finally, for all $(\ell, \ell') \in [m] \times [m]$, we compute

$$\underbrace{\phi'\left(U_1^{(\ell,\ell')}\right)}_{n \times z^2} \cdot \left(\underbrace{\phi'\left(U_2^{(\ell,\ell')}\right)^\top V}_{z^2 \times d}\right),$$

which, similar as in Eq. (37) also takes

$$\sum_{\ell=1}^m \sum_{\ell'=1}^m O\left(z^2 nd\right) = O\left(\frac{B^4 \log^2(n/\delta)}{\epsilon^4} n^{1+o(1)}\right) \tag{38}$$

time.

In conclusion, to construct all of $\phi'\left(U_1^{(\ell,\ell')}\right), \phi'\left(U_2^{(\ell,\ell')}\right) \in \mathbb{R}^{n\times z^2}$, for all $(\ell,\ell') \in [m]\times[m]$, we need Eq. (36) time; computing all of $\underbrace{\phi'\left(U_2^{(\ell,\ell')}\right)^\top}_{z^2\times n} \underbrace{V}_{n\times d}$, we need Eq. (37) time; computing all of $\underbrace{\phi'\left(U_1^{(\ell,\ell')}\right)}_{n\times z^2} \cdot \left(\underbrace{\phi'\left(U_2^{(\ell,\ell')}\right)^\top V}_{z^2\times d}\right)$, we need Eq. (38) time. Adding them all together, we need

$$O\left(\frac{B^6\log^2(n/\delta)}{\epsilon^4} n^{1+o(1)}\right)$$

time. □

I.5 Main Result

We now state the main result of our attention approximation framework. By combining polynomial sketching, bucketing, and sparse additive decompositions, our algorithm approximates the attention computation to within a small ℓ_∞ error, with high probability. Importantly, it does so without requiring bounded entries or restrictive assumptions on the input matrices. The theorem below guarantees both the correctness and the runtime efficiency of our method, showing that the overall computation can be carried out in almost linear time:

Theorem I.5. *Let everything be defined as in Definition I.1.*

Then, with probability $1-\delta$, we can solve the approximate attention computation (Definition 1.2) by outputting $P \in \mathbb{R}^{n\times d}$ satisfying

$$\left\|P - D^{-1}AV\right\|_\infty \lesssim \epsilon_2 \exp(3B^2)\cdot\|V\|_\infty$$

in $O\left(\frac{B^6\log^2(n/\delta)}{\epsilon^4} n^{1+o(1)}\right)$ time.

Proof. **Proof of correctness.**

We define

$$P := \widetilde{D}^{-1}\widetilde{A}V = \mathrm{diag}\left(\widetilde{A}\mathbf{1}_n\right)^{-1}\widetilde{A}V, \tag{39}$$

where

$$\widetilde{A} := \sum_{\ell=1}^m \sum_{\ell'=1}^m \phi'\left(U_1^{(\ell,\ell')}\right)\phi'\left(U_2^{(\ell,\ell')}\right)^\top. \tag{40}$$

By the triangle inequality (see Fact J.3), we have

$$\begin{aligned}\left\|D^{-1}AV - \widetilde{D}^{-1}\widetilde{A}V\right\|_\infty &= \left\|D^{-1}AV - D^{-1}\widetilde{A}V + D^{-1}\widetilde{A}V - \widetilde{D}^{-1}\widetilde{A}V\right\|_\infty \\ &\le \left\|D^{-1}AV - D^{-1}\widetilde{A}V\right\|_\infty + \left\|D^{-1}\widetilde{A}V - \widetilde{D}^{-1}\widetilde{A}V\right\|_\infty. \end{aligned}\tag{41}$$

Considering the first term $\left\|D^{-1}AV - D^{-1}\widetilde{A}V\right\|_\infty$, we have

$$\begin{aligned}\left\|D^{-1}AV - D^{-1}\widetilde{A}V\right\|_\infty &= \left\|D^{-1}\left(A-\widetilde{A}\right)V\right\|_\infty \\ &\le n\left\|D^{-1}\left(A-\widetilde{A}\right)\right\|_\infty \|V\|_\infty\end{aligned}$$

$$\begin{aligned} &\leq n \left\|D^{-1}\right\|_\infty \left\|A - \widetilde{A}\right\|_\infty \|V\|_\infty \\ &\leq n \left\|D^{-1}\right\|_\infty \epsilon_2 \|V\|_\infty , \end{aligned} \tag{42}$$

where the second step follows from the definition of inner product, the third step follows from the fact that D^{-1} is a diagonal matrix, and the last step follows from Lemma I.2.

Now, we consider $\left\|D^{-1}\right\|_\infty$: we have

$$\begin{aligned} \left\|D^{-1}\right\|_\infty = &\max_i \frac{1}{D_{i,i}} \\ = &\frac{1}{\min_i D_{i,i}} \\ = &\frac{1}{\min_i (\exp(QK^\top/d) \cdot \mathbf{1}_n)_{i,*}} \\ = &\frac{1}{\min_i \sum_{j=1}^n \exp\left(\frac{\langle Q_{i,*}, K_{j,*}\rangle}{d}\right)} \\ \leq &\frac{1}{n \cdot \min_{i,j} \exp\left(\frac{\langle Q_{i,*}, K_{j,*}\rangle}{d}\right)} \\ \leq &\frac{\exp(B^2)}{n}, \end{aligned} \tag{43}$$

where the first step follows from the fact that D^{-1} is a diagonal matrix, the third and fourth steps follow from Definition 1.1, and the last step follows from Definition I.1.

Combining Eq. (42) and Eq. (43), we can get

$$\left\|D^{-1}AV - D^{-1}\widetilde{A}V\right\|_\infty \leq \epsilon_2 \exp\left(B^2\right) \|V\|_\infty . \tag{44}$$

Considering the second term $\left\|D^{-1}\widetilde{A}V - \widetilde{D}^{-1}\widetilde{A}V\right\|_\infty$: we have

$$\begin{aligned} \left\|D^{-1}\widetilde{A}V - \widetilde{D}^{-1}\widetilde{A}V\right\|_\infty = &\left\|\left(D^{-1} - \widetilde{D}^{-1}\right)\widetilde{A}V\right\|_\infty \\ \leq &n \cdot \left\|D^{-1} - \widetilde{D}^{-1}\right\|_\infty \cdot \left\|\widetilde{A}\right\|_\infty \cdot \|V\|_\infty \\ = &n \cdot \max_i \left|\frac{1}{D_{i,i}} - \frac{1}{\widetilde{D}_{i,i}}\right| \cdot \left\|\widetilde{A}\right\|_\infty \cdot \|V\|_\infty \\ = &n \cdot \max_i \left|\frac{D_{i,i} - \widetilde{D}_{i,i}}{D_{i,i} \cdot \widetilde{D}_{i,i}}\right| \cdot \left\|\widetilde{A}\right\|_\infty \cdot \|V\|_\infty \\ \leq &\frac{n \cdot \|D - \widetilde{D}\|_\infty}{\min_i D_{i,i} \cdot \min_i \widetilde{D}_{i,i}} \cdot \left\|\widetilde{A}\right\|_\infty \cdot \|V\|_\infty \\ \leq &\frac{\exp(2B^2) \cdot \|D - \widetilde{D}\|_\infty}{n} \cdot \left\|\widetilde{A}\right\|_\infty \cdot \|V\|_\infty , \end{aligned} \tag{45}$$

where the second step follows from the definition of inner product and D^{-1} is a diagonal matrix, the third step follows from the definition of ℓ_∞ norm, and the sixth step follows from Eq. (43).

Note that

$$\begin{aligned} \left\|D - \widetilde{D}\right\|_\infty = &\left\|\mathrm{diag}(A\mathbf{1}_n) - \mathrm{diag}(\widetilde{A}\mathbf{1}_n)\right\|_\infty \\ = &\left\|\left(A - \widetilde{A}\right)\mathbf{1}_n\right\|_\infty \end{aligned}$$

$$\leq n \left\| A - \widetilde{A} \right\|_\infty \|\mathbf{1}_n\|_\infty$$
$$= n \left\| A - \widetilde{A} \right\|_\infty . \tag{46}$$

Combining Eq. (46) and Eq. (45), we have

$$\left\| D^{-1}\widetilde{A}V - \widetilde{D}^{-1}\widetilde{A}V \right\|_\infty \leq \exp(2B^2) \cdot \|A - \widetilde{A}\|_\infty \cdot \left\| \widetilde{A} \right\|_\infty \cdot \|V\|_\infty . \tag{47}$$

Additionally, combining the reverse triangle inequality (see Fact J.3) and Lemma I.2, we have

$$\|\widetilde{A}\|_\infty \leq \|A\|_\infty + \epsilon_2. \tag{48}$$

In particular, considering $\|A\|_\infty$, we note that

$$\begin{aligned} \|A\|_\infty &= \| \exp(QK^\top/d)\|_\infty \\ &= \exp(\|QK^\top\|_\infty/d) \\ &\leq \exp(B^2), \end{aligned} \tag{49}$$

where the first step follows from Definition 1.1 and the second step follows from the definition of inner product.

Combining Eq. (47), Eq. (48), Eq. (49), and Lemma I.2, we have

$$\left\| D^{-1}\widetilde{A}V - \widetilde{D}^{-1}\widetilde{A}V \right\|_\infty \lesssim \epsilon_2 \exp(3B^2) \cdot \|V\|_\infty . \tag{50}$$

Combining Eq. (41), Eq. (44), and Eq. (50), we have

$$\left\| D^{-1}AV - \widetilde{D}^{-1}\widetilde{A}V \right\|_\infty \lesssim \epsilon_2 \exp(3B^2) \cdot \|V\|_\infty .$$

Proof of Running time.

Our goal is to compute P (see Eq. (39) and Eq. (40)).

By Lemma I.3, we have shown that it takes

$$O\left(\frac{B^6 \log^2(n/\delta)}{\epsilon^4} n^{1+o(1)} \right)$$

time to construct $\phi'\left(U_1^{(\ell,\ell')}\right)$ and $\phi'\left(U_2^{(\ell,\ell')}\right)$, for all $(\ell,\ell') \in [m] \times [m]$.

By Lemma I.4, we have shown that it takes

$$O\left(\frac{B^6 \log^2(n/\delta)}{\epsilon^4} n^{1+o(1)} \right)$$

time to compute

$$\underbrace{\phi'\left(U_1^{(\ell,\ell')}\right)}_{n \times z^2} \cdot \left(\underbrace{\phi'\left(U_2^{(\ell,\ell')}\right)^\top}_{z^2 \times d} V \right),$$

and

$$\underbrace{\phi'\left(U_2^{(\ell,\ell')}\right)^\top}_{z^2 \times n} \underbrace{V}_{n \times d},$$

for all $(\ell, \ell') \in [m] \times [m]$.

Similarly, it also takes $O\left(\frac{B^6 \log^2(n/\delta)}{\epsilon^4} n^{1+o(1)}\right)$ time to compute

$$\underbrace{\phi'\left(U_1^{(\ell,\ell')}\right)}_{n \times z^2} \cdot \left(\underbrace{\phi'\left(U_2^{(\ell,\ell')}\right)^\top \mathbf{1}_n}_{z^2 \times 1} \right),$$

and

$$\underbrace{\phi'\left(U_2^{(\ell,\ell')}\right)^\top}_{z^2 \times n} \underbrace{\mathbf{1}_n}_{n \times 1},$$

for all $(\ell, \ell') \in [m] \times [m]$.

This implies that we can compute

$$\begin{aligned} \widetilde{D}^{-1} = & \operatorname{diag}\left(\widetilde{A}\mathbf{1}_n\right)^{-1} \\ = & \operatorname{diag}\left(\sum_{\ell=1}^m \sum_{\ell'=1}^m \phi'\left(U_1^{(\ell,\ell')}\right) \phi'\left(U_2^{(\ell,\ell')}\right)^\top \mathbf{1}_n\right)^{-1} \end{aligned}$$

in $O\left(\frac{B^6 \log^2(n/\delta)}{\epsilon^4} n^{1+o(1)}\right)$ time.

Finally, as $\widetilde{D}^{-1}$ is a diagonal matrix, computing

$$P = \widetilde{D}^{-1}\widetilde{A}V = \sum_{\ell=1}^m \sum_{\ell'=1}^m \left(\underbrace{\widetilde{D}^{-1}}_{n \times n} \underbrace{\phi'\left(U_1^{(\ell,\ell')}\right)}_{n \times z^2} \right) \cdot \left(\underbrace{\phi'\left(U_2^{(\ell,\ell')}\right)^\top}_{z^2 \times n} \underbrace{V}_{n \times d} \right)$$

takes $O\left(\frac{B^6 \log^2(n/\delta)}{\epsilon^4} n^{1+o(1)}\right)$ time.

□

J MORE FACTS

In this section, we present basic mathematical properties that support our theoretical proofs.

Fact J.1. *Let $A, B \in \mathbb{R}^{n \times n}$.*

Then, we have

$$|\operatorname{supp}(A+B)| \leq |\operatorname{supp}(A)| + |\operatorname{supp}(B)|.$$

Proof. It suffices to show that

$$\operatorname{supp}(A+B) \subseteq \operatorname{supp}(A) \cup \operatorname{supp}(B). \tag{51}$$

Suppose (i, j) is an arbitrary element in $\operatorname{supp}(A+B)$.

Then, we have $(A+B)_{i,j} \neq 0$.

Note that by the basic definition of matrix addition, we have

$$(A+B)_{i,j} = A_{i,j} + B_{i,j}$$

$$\neq 0,$$

which implies that either $A_{i,j} \neq 0$ or $B_{i,j} \neq 0$.

Therefore, $(i,j) \in \operatorname{supp}(A)$ or $(i,j) \in \operatorname{supp}(B)$, which completes the proof of Eq. (51). □

Fact J.2. *Let $\{A^{(\ell,\ell')}\}_{\ell,\ell' \in [m]}$ be a collection of $n \times n$ matrices forming a support basis of $A \in \mathbb{R}^{n \times n}$.*

Then, we have

$$\sum_{\ell=1}^{m} \sum_{\ell'=1}^{m} \left| \operatorname{supp}\left(A^{(\ell,\ell')}\right) \right| = |\operatorname{supp}(A)| \leq n^2.$$

Proof. By the definition of support basis (see Definition B.5), we know that $A^{(\ell,\ell')}$'s are disjoint matrices (Definition B.1).

Therefore, we have

$$\begin{aligned} \operatorname{supp}(A) = & \operatorname{supp}\left(\sum_{\ell,\ell'} A^{(\ell,\ell')}\right) \\ = & \bigcup_{\ell,\ell'} \operatorname{supp}\left(A^{(\ell,\ell')}\right). \end{aligned}$$

Thus, we can further get

$$\begin{aligned} |\operatorname{supp}(A)| = & \left| \bigcup_{\ell,\ell'} \operatorname{supp}\left(A^{(\ell,\ell')}\right) \right| \\ = & \sum_{\ell,\ell'} \left| \operatorname{supp}\left(A^{(\ell,\ell')}\right) \right|. \end{aligned}$$

Finally, since $A \in \mathbb{R}^{n \times n}$, we can get that $|\operatorname{supp}(A)| \leq n^2$. □

Fact J.3. *Let $u, v \in \mathbb{R}^n$.*

Then, we have

- $|\langle u, v \rangle| \leq \|u\|_2 \cdot \|v\|_2$ *(Cauchy–Schwarz inequality).*
- $\|u\|_2 \leq \|u\|_1$.
- $\langle u, v \rangle = u^\top v$.
- $\langle u, u \rangle = u^\top u = \|u\|_2^2$.
- $\|u + v\|_2 \leq \|u\|_2 + \|v\|_2$ *(triangle inequality).*
- $|\|u\|_2 - \|v\|_2| \leq \|u - v\|_2$ *(reverse triangle inequality).*

The (reverse) triangle inequality holds in all metric spaces. In particular, it applies to $(\mathbb{R}, |\cdot|)$, $(\mathbb{R}^{n \times d}, \|\cdot\|_p)$, and $(\mathbb{R}^n, \|\cdot\|_p)$, for all positive integers n, d, and $p \in \{1, 2, \ldots, \infty\}$.

Fact J.4 (Integral analysis)**.** *We have*

- **Part 1.** *if x is the variable, then*

$$\begin{aligned} & \int_1^{m+1} \frac{b^4 (1+\epsilon)^{2x+2y}}{\log^2 n} n^{\frac{cb^2(1+\epsilon)^{x+y}}{\log n}} \, \mathrm{d}x \\ = & \frac{cb^2 (1+\epsilon)^{y+m+1} e^{b^2 c(\epsilon+1)^{y+m+1}} - cb^2 (1+\epsilon)^{y+1} e^{b^2 c(\epsilon+1)^{y+1}} - e^{b^2 c(\epsilon+1)^{y+m+1}} + e^{b^2 c(\epsilon+1)^{y+1}}}{c^2 \log^2 n \log(\epsilon+1)} \end{aligned}$$

- **Part 2.** *if y is the variable, then*

$$\int_1^{m+1} \frac{b^2 (1+\epsilon)^{y+m+1} e^{b^2 c(\epsilon+1)^{y+m+1}}}{c \log^2(n) \log(\epsilon+1)} \mathrm{d}y = \frac{e^{b^2 c(\epsilon+1)^{2m+2}} - e^{b^2 c(\epsilon+1)^{m+2}}}{c^2 \log^2(n) \log^2(\epsilon+1)}.$$

Proof. **Proof of Part 1.**

By the linearity of the integral, we have

$$\int_1^{m+1} \frac{b^4 (1+\epsilon)^{2x+2y}}{\log^2 n} n^{\frac{cb^2(1+\epsilon)^{x+y}}{\log n}} \mathrm{d}x = \frac{b^4 (1+\epsilon)^{2y}}{\log^2 n} \int_1^{m+1} (1+\epsilon)^{2x} n^{\frac{cb^2(1+\epsilon)^{x+y}}{\log n}} \mathrm{d}x. \tag{52}$$

We can further get

$$\int_1^{m+1} (\epsilon+1)^{2x} n^{\frac{b^2 c(\epsilon+1)^{x+y}}{\log(n)}} \mathrm{d}x = \int_1^{m+1} (\epsilon+1)^x \log(\epsilon+1) \cdot \frac{(\epsilon+1)^x e^{b^2 c(\epsilon+1)^{x+y}}}{\log(\epsilon+1)} \mathrm{d}x$$

Now, we use u-substitution by letting $u = (\epsilon+1)^x$, which implies that

$$x = \frac{\log(u)}{\log(\epsilon+1)}$$

and

$$\mathrm{d}x = \frac{1}{(\epsilon+1)^x \log(\epsilon+1)} \mathrm{d}u$$

Therefore, we have

$$\int_1^{m+1} (\epsilon+1)^{2x} n^{\frac{b^2 c(\epsilon+1)^{x+y}}{\log(n)}} \mathrm{d}x = \frac{1}{\log(\epsilon+1)} \int_{1+\epsilon}^{(1+\epsilon)^{m+1}} u e^{b^2 c(\epsilon+1)^y u} \mathrm{d}u \tag{53}$$

Now, we consider

$$\int_{1+\epsilon}^{(1+\epsilon)^{m+1}} u e^{b^2 c(\epsilon+1)^y u} \mathrm{d}u$$

We use integration by parts, namely $\int fg' = fg - \int f'g$.

With

$$f = u, \quad g' = e^{b^2 c(\epsilon+1)^y u}$$
$$f' = 1, \quad g = \frac{e^{b^2 c(\epsilon+1)^y u}}{b^2 c(\epsilon+1)^y},$$

we have

$$\begin{aligned}
&\int_{1+\epsilon}^{(1+\epsilon)^{m+1}} u e^{b^2 c(\epsilon+1)^y u} \mathrm{d}u \\
=\ & \frac{u e^{b^2 c(\epsilon+1)^y u}}{b^2 c(\epsilon+1)^y} \Bigg|_{1+\epsilon}^{(1+\epsilon)^{m+1}} - \int_{1+\epsilon}^{(1+\epsilon)^{m+1}} \frac{e^{b^2 c(\epsilon+1)^y u}}{b^2 c(\epsilon+1)^y} \mathrm{d}u \\
=\ & \left(\frac{(1+\epsilon)^{m+1} e^{b^2 c(\epsilon+1)^{y+m+1}} - (1+\epsilon) e^{b^2 c(\epsilon+1)^{y+1}}}{b^2 c(\epsilon+1)^y} \right) - \int_{1+\epsilon}^{(1+\epsilon)^{m+1}} \frac{e^{b^2 c(\epsilon+1)^y u}}{b^2 c(\epsilon+1)^y} \mathrm{d}u.
\end{aligned} \tag{54}$$

We again use u-substitution by letting $v = b^2 c(\epsilon+1)^y u$ so that

$$\mathrm{d}u = \frac{1}{b^2 c(\epsilon+1)^y} \mathrm{d}v$$

Therefore, we get

$$\int_{1+\epsilon}^{(1+\epsilon)^{m+1}} \frac{e^{b^2c(\epsilon+1)^y u}}{b^2c(\epsilon+1)^y}\,\mathrm{d}u = \frac{1}{b^4c^2(\epsilon+1)^{2y}} \int_{b^2c(\epsilon+1)^y(1+\epsilon)}^{b^2c(\epsilon+1)^y(1+\epsilon)^{m+1}} e^v\,\mathrm{d}v$$
$$= \frac{e^{b^2c(\epsilon+1)^y(1+\epsilon)^{m+1}} - e^{b^2c(\epsilon+1)^y(1+\epsilon)}}{b^4c^2(\epsilon+1)^{2y}}. \tag{55}$$

Combining Eq. (52), (53), (54), and (55), we have

$$\int_1^{m+1} \frac{b^4\,(1+\epsilon)^{2x+2y}}{\log^2 n} n^{\frac{cb^2(1+\epsilon)^{x+y}}{\log n}}\,\mathrm{d}x$$
$$= \left(\frac{b^4\,(1+\epsilon)^{2y}}{\log^2 n}\right) \cdot \left(\frac{\frac{(1+\epsilon)^{m+1}e^{b^2c(\epsilon+1)^{y+m+1}} - (1+\epsilon)e^{b^2c(\epsilon+1)^{y+1}}}{b^2c(\epsilon+1)^y} - \frac{e^{b^2c(\epsilon+1)^y(1+\epsilon)^{m+1}} - e^{b^2c(\epsilon+1)^y(1+\epsilon)}}{b^4c^2(\epsilon+1)^{2y}}}{\log(\epsilon+1)}\right)$$
$$= \frac{\frac{b^2(1+\epsilon)^y(1+\epsilon)^{m+1}e^{b^2c(\epsilon+1)^{y+m+1}} - b^2(1+\epsilon)^y(1+\epsilon)e^{b^2c(\epsilon+1)^{y+1}}}{c} - \frac{e^{b^2c(\epsilon+1)^y(1+\epsilon)^{m+1}} - e^{b^2c(\epsilon+1)^y(1+\epsilon)}}{c^2}}{\log^2 n \log(\epsilon+1)}$$
$$= \frac{cb^2\,(1+\epsilon)^{y+m+1}\,e^{b^2c(\epsilon+1)^{y+m+1}} - cb^2\,(1+\epsilon)^{y+1}\,e^{b^2c(\epsilon+1)^{y+1}} - e^{b^2c(\epsilon+1)^{y+m+1}} + e^{b^2c(\epsilon+1)^{y+1}}}{c^2 \log^2 n \log(\epsilon+1)}.$$

Proof of Part 2.

Now, we consider

$$\int_1^{m+1} \frac{b^2(\epsilon+1)^{y+m+1}e^{b^2c(\epsilon+1)^{y+m+1}}}{c\log^2(n)\log(\epsilon+1)}\,\mathrm{d}y$$

We use u-substitution by defining $u = b^2c(\epsilon+1)^{y+m+1}$, which implies

$$\mathrm{d}y = \frac{1}{b^2c(\epsilon+1)^{y+m+1}\log(\epsilon+1)}\mathrm{d}u.$$

Therefore, we have

$$\int_1^{m+1} \frac{b^2(\epsilon+1)^{y+m+1}e^{b^2c(\epsilon+1)^{y+m+1}}}{c\log^2(n)\log(\epsilon+1)}\,\mathrm{d}y$$
$$= \frac{1}{c\log^2(n)\log(\epsilon+1)} \int_1^{m+1} b^2(\epsilon+1)^{y+m+1}e^{b^2c(\epsilon+1)^{y+m+1}}\,\mathrm{d}y$$
$$= \frac{1}{c^2\log^2(n)\log^2(\epsilon+1)} \int_{b^2c(\epsilon+1)^{m+2}}^{b^2c(\epsilon+1)^{2m+2}} e^u\,\mathrm{d}u$$
$$= \frac{e^{b^2c(\epsilon+1)^{2m+2}} - e^{b^2c(\epsilon+1)^{m+2}}}{c^2\log^2(n)\log^2(\epsilon+1)}.$$

□

Fact J.5 (Multiplicative Chernoff Bound)**.** *Let $X_1, X_2, \ldots, X_n$ be independent random variables taking values in $[0,1]$, and let $X = \sum_{i=1}^n X_i$ with $\mu = \mathbb{E}[X]$. Then for all $\delta > 0$, the following holds:*

$$\Pr[X \geq (1+\delta)\mu] \leq \exp\left(-\frac{\delta^2\mu}{3}\right) \quad \textit{for } 0 < \delta \leq 1,$$
$$\Pr[X \geq (1+\delta)\mu] \leq \exp\left(-\frac{\delta\mu}{3}\right) \quad \textit{for } \delta > 1.$$

Fact J.6. *Let $a, b \in \mathbb{R}^d$. Let p be an arbitrary positive integer.*

Then, we have

- **Part 1.** $\langle a^{\otimes 2}, b^{\otimes 2} \rangle = \langle a, b \rangle^2$.
- **Part 2.** $\langle a^{\otimes p}, b^{\otimes p} \rangle = \langle a, b \rangle^p$ *(generalizing* **Part 1** *to arbitrary p).*
- **Part 3.** $\|a^{\otimes p}\|_2 = \|a\|_2^p$.

Proof. **Proof of Part 1.**

We have

$$\begin{aligned}
\langle a^{\otimes 2}, b^{\otimes 2} \rangle &= \langle a \otimes a, b \otimes b \rangle \\
&= \sum_{i=1}^d \sum_{j=1}^d (a_i a_j)(b_i b_j) \\
&= \sum_{i=1}^d \sum_{j=1}^d a_i b_i a_j b_j \\
&= \left(\sum_{i=1}^d a_i b_i \right) \left(\sum_{j=1}^d a_j b_j \right) \\
&= \left(\sum_{i=1}^d a_i b_i \right)^2 \\
&= \langle a, b \rangle^2,
\end{aligned}$$

where the first step follows from the definition of $a^{\otimes 2}$ and $b^{\otimes 2}$, the second step follows from the definition of inner product and Kronecker product, the third step follows from the associative law of multiplication, and the last step follows from the definition of inner product.

Proof of Part 2

We prove this part using mathematical induction.

We treat **Part 1** as the base case, and below, we prove the inductive case.

Assume for all given positive integer k, we have

$$\langle a^{\otimes k}, b^{\otimes k} \rangle = \langle a, b \rangle^k. \tag{56}$$

Similar as what we have in **Part 1**, we can get

$$\begin{aligned}
\langle a^{\otimes (k+1)}, b^{\otimes (k+1)} \rangle &= \langle a^{\otimes k} \otimes a, b^{\otimes k} \otimes b \rangle \\
&= \sum_{j=1}^d \sum_{i=1}^{d^k} \left(\left(a^{\otimes k} \right)_i a_j \right) \left(\left(b^{\otimes k} \right)_i b_j \right) \\
&= \sum_{j=1}^d \sum_{i=1}^{d^k} \left(a^{\otimes k} \right)_i a_j \left(b^{\otimes k} \right)_i b_j \\
&= \left(\sum_{i=1}^{d^k} \left(a^{\otimes k} \right)_i \left(b^{\otimes k} \right)_i \right) \left(\sum_{j=1}^d a_j b_j \right) \\
&= \langle a^{\otimes k}, b^{\otimes k} \rangle \cdot \langle a, b \rangle,
\end{aligned}$$

where the first step follows from the definition of $a^{\otimes(k+1)}$ and $b^{\otimes(k+1)}$, the second step follows from the definition of inner product and Kronecker product, the third step follows from the associative law of multiplication, and the last step follows from the definition of inner product.

By the inductive hypothesis (Eq. (56)), we have

$$\begin{aligned}\langle a^{\otimes(k+1)}, b^{\otimes(k+1)}\rangle &= \langle a, b\rangle^k \cdot \langle a, b\rangle \\ &= \langle a, b\rangle^{k+1}.\end{aligned}$$

Proof of Part 3.

We can get

$$\begin{aligned}\|a^{\otimes p}\|_2^2 &= \sum_{i_1=1}^{d} \cdots \sum_{i_p=1}^{d} (a_{i_1} a_{i_2} \cdots a_{i_p})^2 \\ &= \sum_{i_1=1}^{d} \cdots \sum_{i_p=1}^{d} a_{i_1}^2 a_{i_2}^2 \cdots a_{i_p}^2 \\ &= \left(\sum_{i=1}^{d} a_i^2\right)^p \\ &= \|a\|_2^{2p},\end{aligned}$$

where the first step follows from the definition of the ℓ_2 norm and the Kronecker product, and the last step follows from the definition of the ℓ_2 norm.

Taking square roots of both sides, we have

$$\|a^{\otimes p}\|_2 = \|a\|_2^p$$

□

Fact J.7 (Markov's Inequality)**.** *Let X be a non-negative random variable and let $a > 0$. Then*

$$\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}.$$

Fact J.8 (Union Bound)**.** *Let $A_1, A_2, \ldots, A_n$ be events in a probability space. Then*

$$\Pr\left[\bigcup_{i=1}^{n} A_i\right] \leq \sum_{i=1}^{n} \Pr[A_i].$$

Fact J.9. *Let $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times n}$ be disjoint matrices. Let p and k be arbitrary positive integers. Then, we can get*

- **Part 1.** $(A+B)^{\circ p} = (A)^{\circ p} + (B)^{\circ p}$,
- **Part 2** $\|A+B\|_p^p = \|A\|_p^p + \|B\|_p^p$,
- **Part 3** $\left\|A^{\circ k}\right\|_p^p \leq \|A\|_p^{pk}$.

Proof. **Proof of Part 1.**

Since $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times n}$ are disjoint matrices, for all arbitrary $(i,j) \in [n] \times [n]$, we can either have

$$A_{i,j} \neq 0 \text{ and } B_{i,j} = 0 \tag{57}$$

or

$$B_{i,j} \neq 0 \text{ and } A_{i,j} = 0. \tag{58}$$

If Eq. (57) holds, then we can get

$$\begin{aligned}\left((A+B)^{\circ p}\right)_{i,j} = &\ (A+B)_{i,j}^p \\ = &\ A_{i,j}^p.\end{aligned}$$

If Eq. (58) holds, then we can get

$$\begin{aligned}\left((A+B)^{\circ p}\right)_{i,j} = &\ (A+B)_{i,j}^p \\ = &\ B_{i,j}^p.\end{aligned}$$

Proof of Part 2.

We can get

$$\begin{aligned}\|A+B\|_p^p = &\ \sum_{i,j}(A+B)_{i,j}^p \\ = &\ \sum_{i,j}(A_{i,j}^p + B_{i,j}^p) \\ = &\ \sum_{i,j} A_{i,j}^p + \sum_{i,j} B_{i,j}^p \\ = &\ \|A\|_p^p + \|B\|_p^p,\end{aligned}$$

where the first step follows from the definition of ℓ_p norm and the second step follows from **Part 1**.

Proof of Part 3.

We have

$$\begin{aligned}\left\|A^{\circ k}\right\|_p^p = &\ \sum_{i,j}(A^{\circ k})_{i,j}^p \\ = &\ \sum_{i,j}(A_{i,j}^k)^p \\ = &\ \sum_{i,j}(A_{i,j}^p)^k \\ \leq &\ \left(\sum_{i,j} A_{i,j}^p\right)^k \\ = &\ \|A\|_p^{pk}.\end{aligned}$$

□

K MORE RELATED WORK

Attention regression problems Another line of work that attempts to reduce the computational complexity of attention approximation transforms the attention computation into regression problems and applies the approximate Newton method to solve them. These methods use sketching matrices to reduce the dimensionality of the Hessian in the Newton method, thereby accelerating the algorithm with provable guarantees. However, most of these works simplify the attention computation problem (Definition 1.1).

For example, Li et al. (2025) proposes a softmax regression formulation,

$$\min_{x\in\mathbb{R}^d} \left\|\langle \exp(Ax), \mathbf{1}_n\rangle^{-1} \exp(Ax) - c\right\|_2^2$$

where the value matrix is completely ignored. Gao et al. (2025b) studies a rescaled version of softmax regression,

$$\min_{x\in\mathbb{R}^d} \|\exp(Ax) - \langle \exp(Ax), \mathbf{1}_n\rangle c\|_2^2$$

Song et al. (2024) analyzes an exponential regression problem,

$$\min_{x \in \mathbb{R}^n} \left\| \exp\left(AA^\top\right) x - b \right\|_2^2,$$

where both D^{-1} and V are omitted. Gao et al. (2025a) is the only work that does not make any such simplifications, studying the attention regression problem

$$\min_{X, Y \in \mathbb{R}^{d \times d}} \left\| D\left(X\right)^{-1} \exp\left(A_1 X A_2^\top\right) A_3 Y - B \right\|_F^2.$$

However, the approximate Newton method can only solve convex problems, whereas attention computation is inherently non-convex. To apply the approximate Newton method, Gao et al. (2025a) still relies on regularization terms and the choice of a good initialization point for the iterative procedure—assumptions that limit the applicability of the proposed algorithm.

Sketching For a large data matrix $A \in \mathbb{R}^{n \times d}$, earlier works have centered on designing a sketching matrix $S \in \mathbb{R}^{m \times n}$ with $m \ll n$ such that, for every $x \in \mathbb{R}^d$, $\|SAx\|_2^2 = (1 \pm \epsilon)\|Ax\|_2^2$, where $\epsilon \in (0, 1)$ denotes the approximation error. A matrix S with this property is known as a subspace embedding for A. This idea has become a standard tool across many machine learning tasks, including linear regression (Price et al., 2017; Song et al., 2023b,c), (weighted) low-rank approximation (Clarkson and Woodruff, 2017; Song et al., 2025b), tensor power method (Deng et al., 2023b), online weighted matching problem (Song et al., 2025a), low-rank matrix completion (Gu et al., 2024), k-means clustering (Liang et al., 2022), and attention mechanisms (Kacham et al., 2024).

Large language models Besides the line of works that focus on the computational efficiency of LLMs, such as Liang et al. (2024b); Chen et al. (2025a); Liang et al. (2025), there is another line of work on memory efficiency, such as LoRA (Hu et al., 2022), GaLore (Zhao et al., 2024), SARA (Zhang et al., 2025a), KV-Cache compression (Chen et al., 2025d), and CoVE (Zhang et al., 2025b). Another line of work focuses on analyzing the softmax unit in attention, such as binary hypothesis testing (Gu et al., 2025), softmax regression (Li et al., 2025; Song et al., 2023a; Li et al., 2023). Other works focus on the circuit complexity of LLMs (Ke et al., 2025; Chen et al., 2025b), limitations of LLMs (Chen et al., 2025c), and hallucination (Lin et al., 2025). In reinforcement learning (Zhang et al., 2025c, 2023a), Wang et al. (2024) design reinforcement learning targeted attack on LLMs, and Peiyuan et al. (2024) propose the reinforcement learning framework of LLM agents. Furthermore, the application of large transformers with long-context capabilities is not limited to language processing; broader domains such as time-series prediction (Zhang et al., 2023b, 2022a; Jin et al., 2023; Liu et al., 2024; Gruver et al., 2023) are also of practical importance. Extending and validating our method to preserve the accuracy of large transformer models on these tasks is, therefore, a worthwhile direction for future work.

L QUERY AND KEY ENTRIES DISTRIBUTION

In this section, we empirically validate our assumption that the entries of query and key matrices resemble sub-Gaussian distributions on transformer architectures such as TinyLlama-1.1B (Zhang et al., 2024), LLaDA-8B-Base (Nie et al., 2025), OPT-1.3B (Zhang et al., 2022b), and Phi-2 (Javaheripi et al., 2023) across multiple layers. Their model IDs are GSAI-ML/LLaDA-8B-Base (LLaDA-8B-Base), facebook/opt-1.3b (OPT-1.3B), TinyLlama/TinyLlama-1.1B-Chat-v1.0 (TinyLlama-1.1B), and microsoft/phi-2 (Phi-2).

To empirically validate our assumption, we visualize the distribution of the entries of Q and K across multiple layers, generated by natural human-like tokens (Appendix L.2, L.3, L.4, and L.5). We observe that for all of the layers of attention in these transformer architectures (TinyLlama-1.1B (Zhang et al., 2024), LLaDA-8B-Base (Nie et al., 2025), OPT-1.3B (Zhang et al., 2022b), and Phi-2 (Javaheripi et al., 2023)), the entry distributions are sub-Gaussian-like, which satisfies the "practical assumption" made in this paper. In Appendix L.1 we present the entry distribution of Q and K of TinyLlama-1.1B (Zhang et al., 2024) by deliberately constructing a non-human-like token sequence consisting of the character "h" repeated 2048 times. For these tokens, the distributions of the entries of Q and K are more dispersed in the early layers (layers 0–2, Figures 5–7). However, starting from layer 3 onward, all entries of Q and K become increasingly concentrated (Figures 8–26). LLMs are trained on natural human-like tokens, so it is expected that the entries of Q and K remain tightly concentrated

across all layers. Our result in Appendix L.1 can be viewed as a worst-case scenario: even if the entries are not well centered around the mean in the first few iterations, they become more tightly concentrated as ℓ increases.

In Appendix L.2, we show that in TinyLlama-1.1B (Zhang et al., 2024), after the first few layers, the entries of Q and K become tightly concentrated with light tails. We present the entry distribution of Q and K of TinyLlama-1.1B (Zhang et al., 2024) from layer 0 to layer 21. In Appendix L.3, we present the entry distribution of Q and K of OPT-1.3B (Zhang et al., 2022b) from layer 0 to layer 23. In Appendix L.4, we present the entry distribution of Q and K of LLaDA-8B-Base (Nie et al., 2025) from layer 0 to layer 31. In Appendix L.5, we present the entry distribution of Q and K of Phi-2 (Javaheripi et al., 2023) from layer 0 to layer 31.

L.1 TinyLlama-1.1B With Repeated Token

In this section, we present the entry distribution of Q and K in TinyLlama-1.1B (Zhang et al., 2024) from layer 0 to layer 21. The token sequence consists of the character "h" repeated 2048 times.

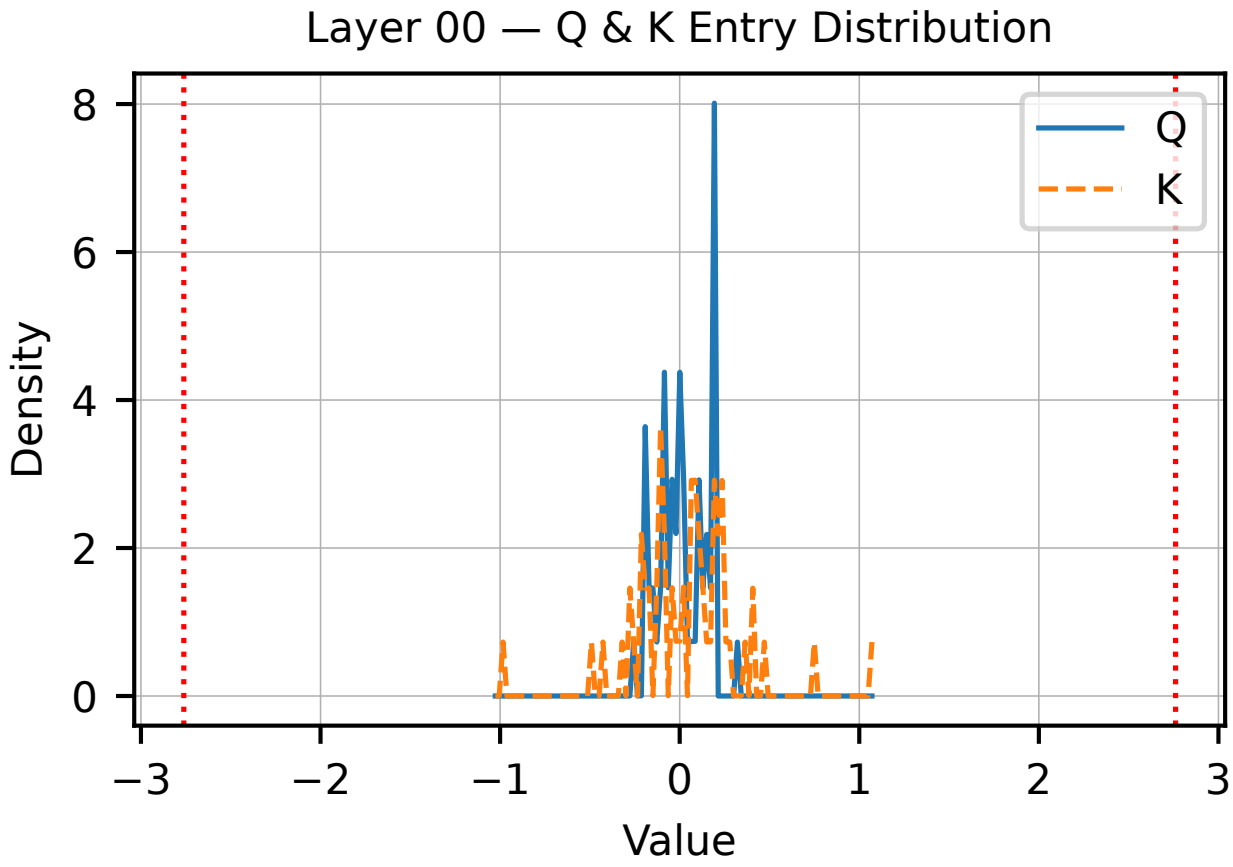


Figure 5: Distribution of entries in the query and key matrices of Layer 0 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

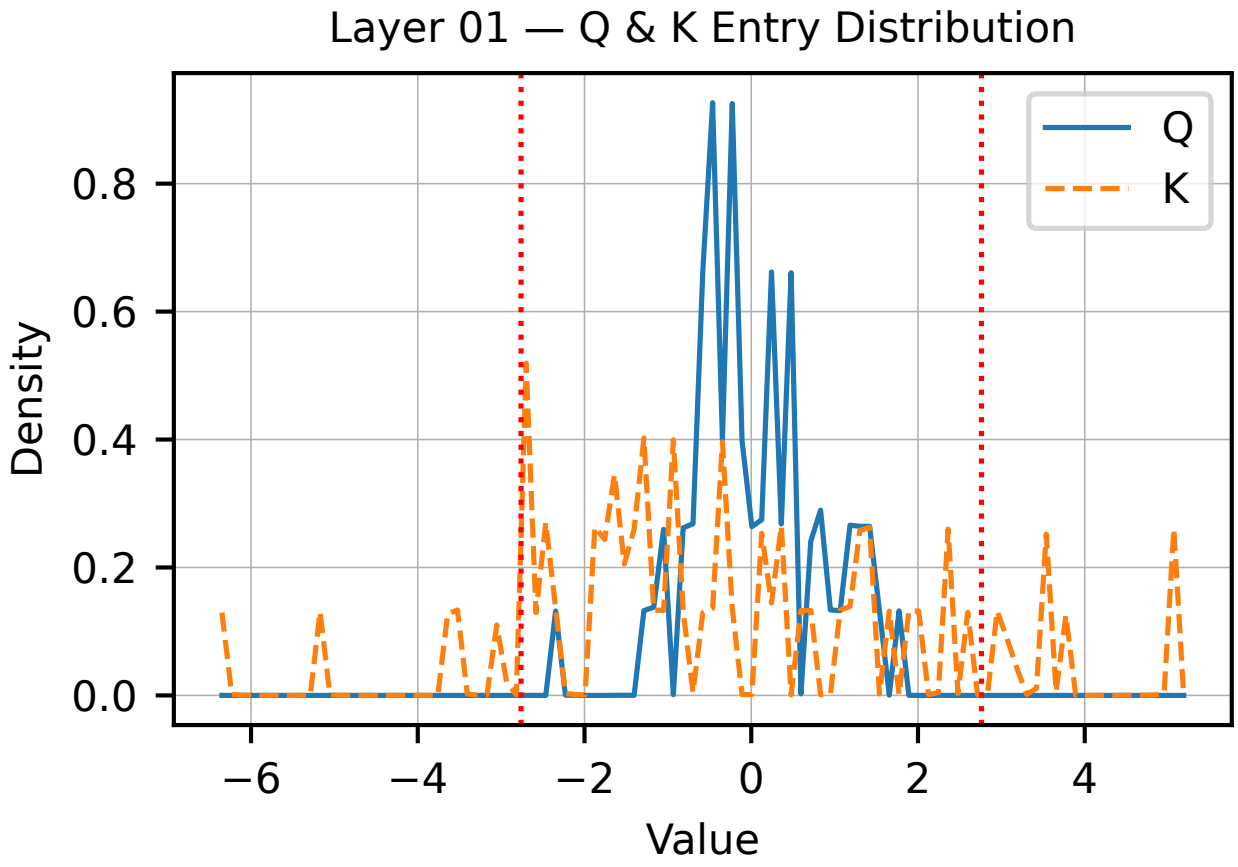


Figure 6: Distribution of entries in the query and key matrices of Layer 1 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

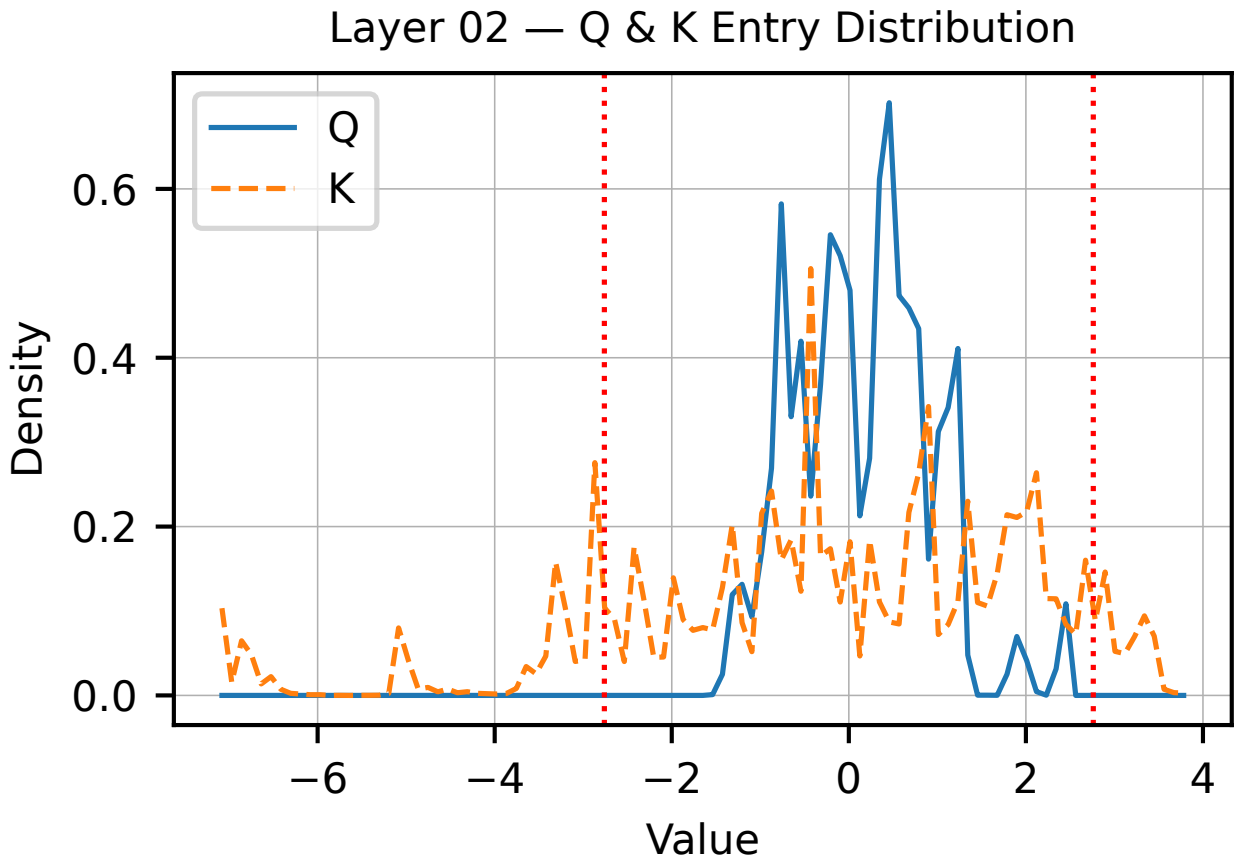


Figure 7: Distribution of entries in the query and key matrices of Layer 2 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

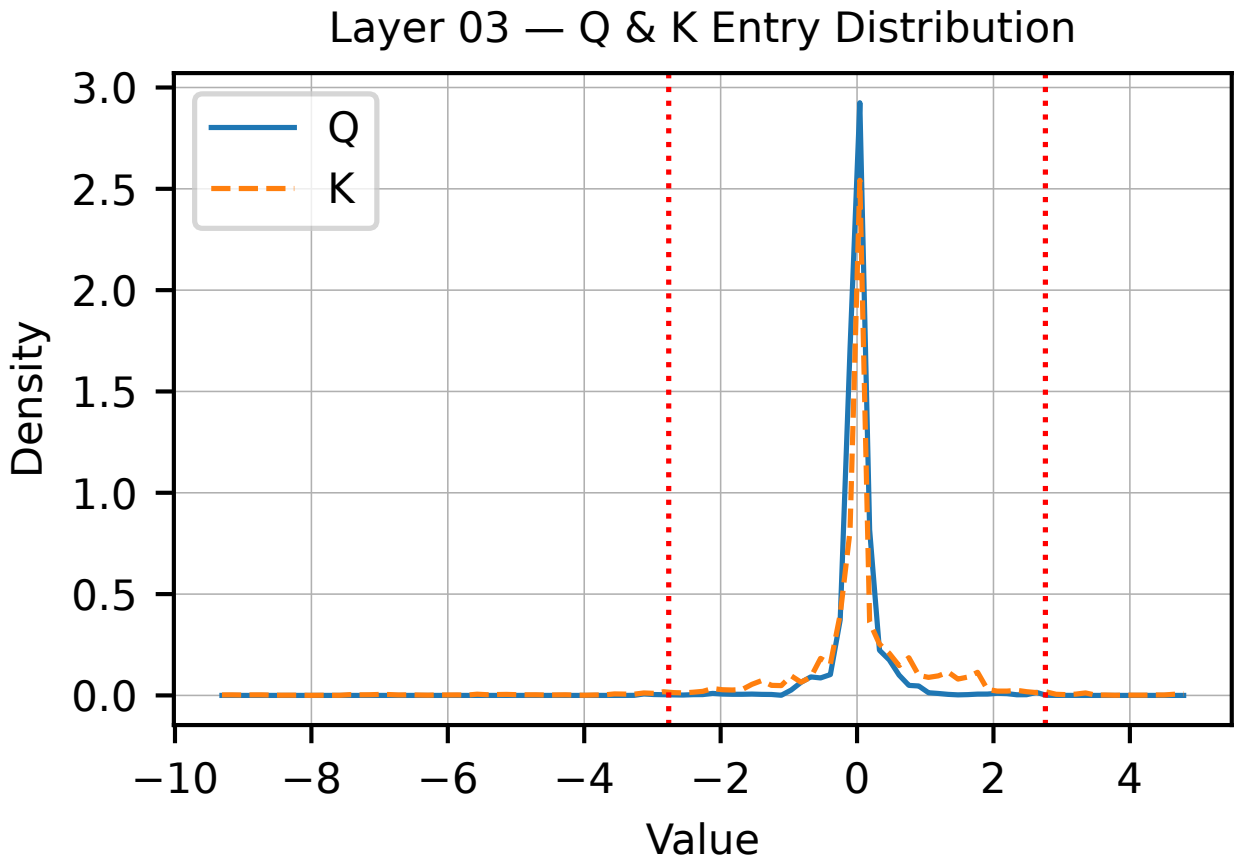


Figure 8: Distribution of entries in the query and key matrices of Layer 3 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

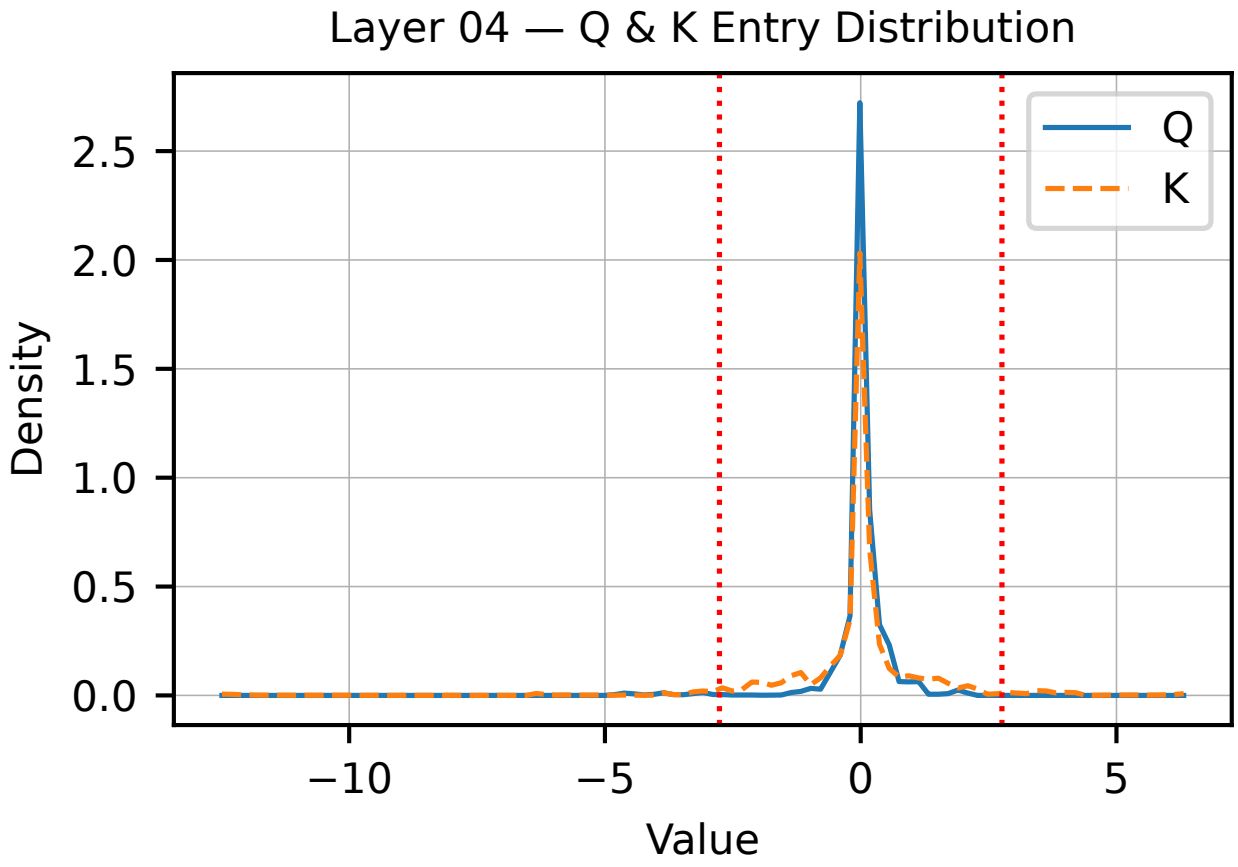


Figure 9: Distribution of entries in the query and key matrices of Layer 4 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

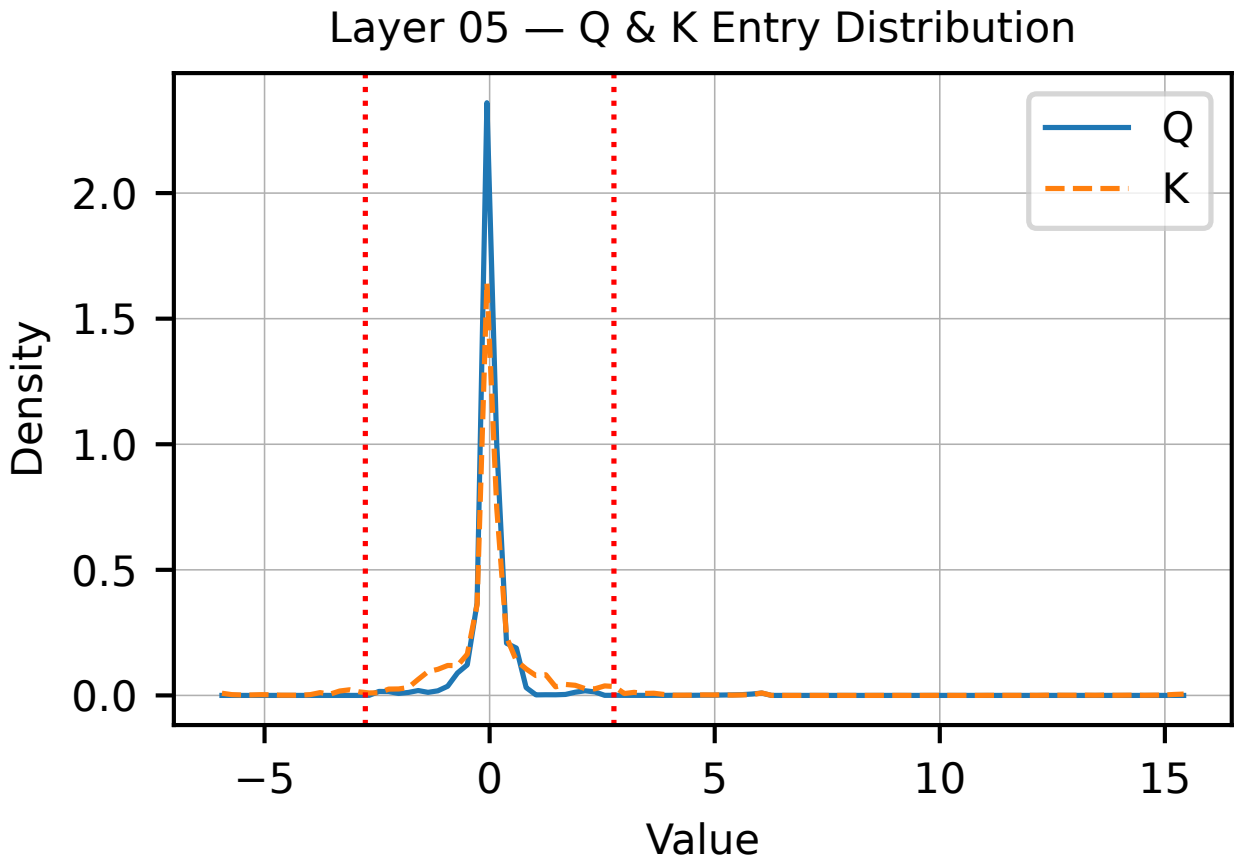


Figure 10: Distribution of entries in the query and key matrices of Layer 5 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

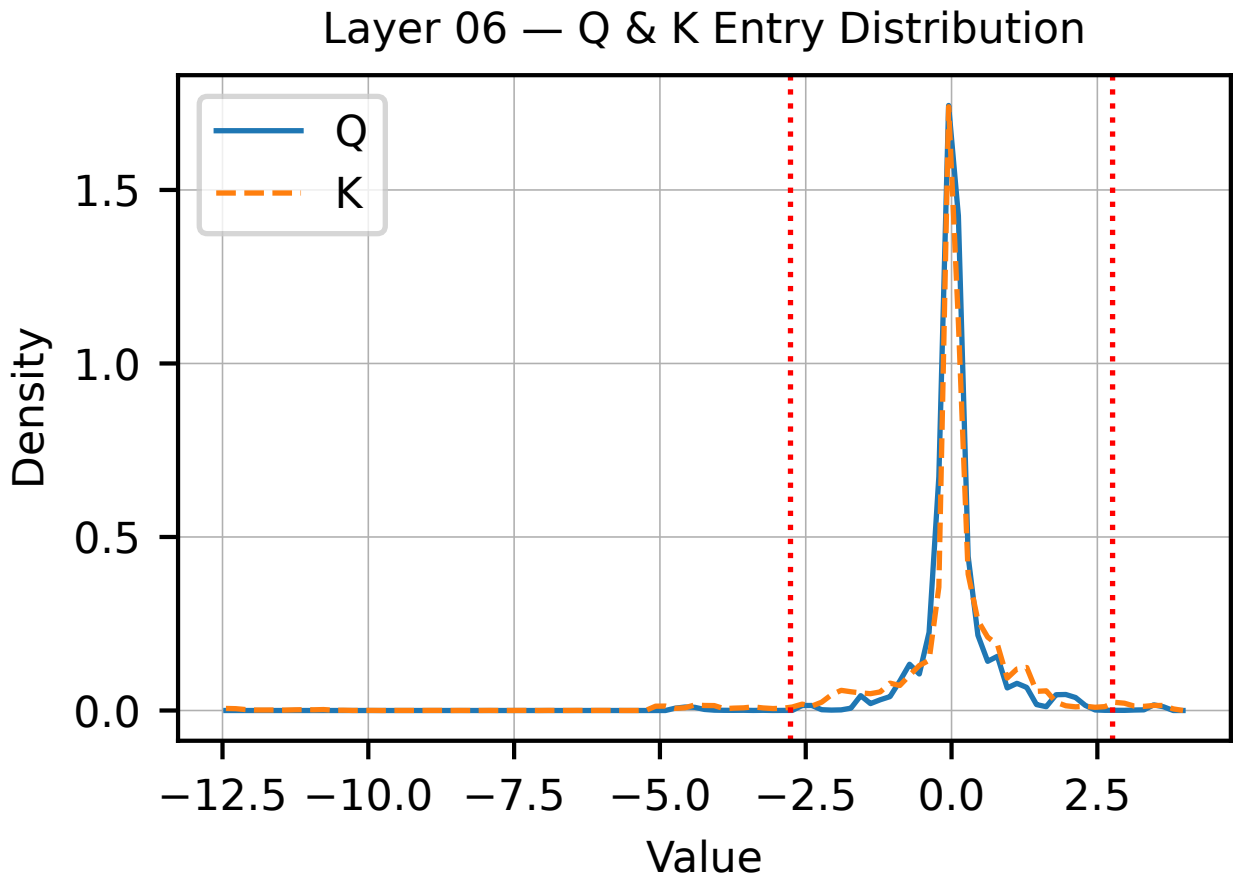


Figure 11: Distribution of entries in the query and key matrices of Layer 6 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

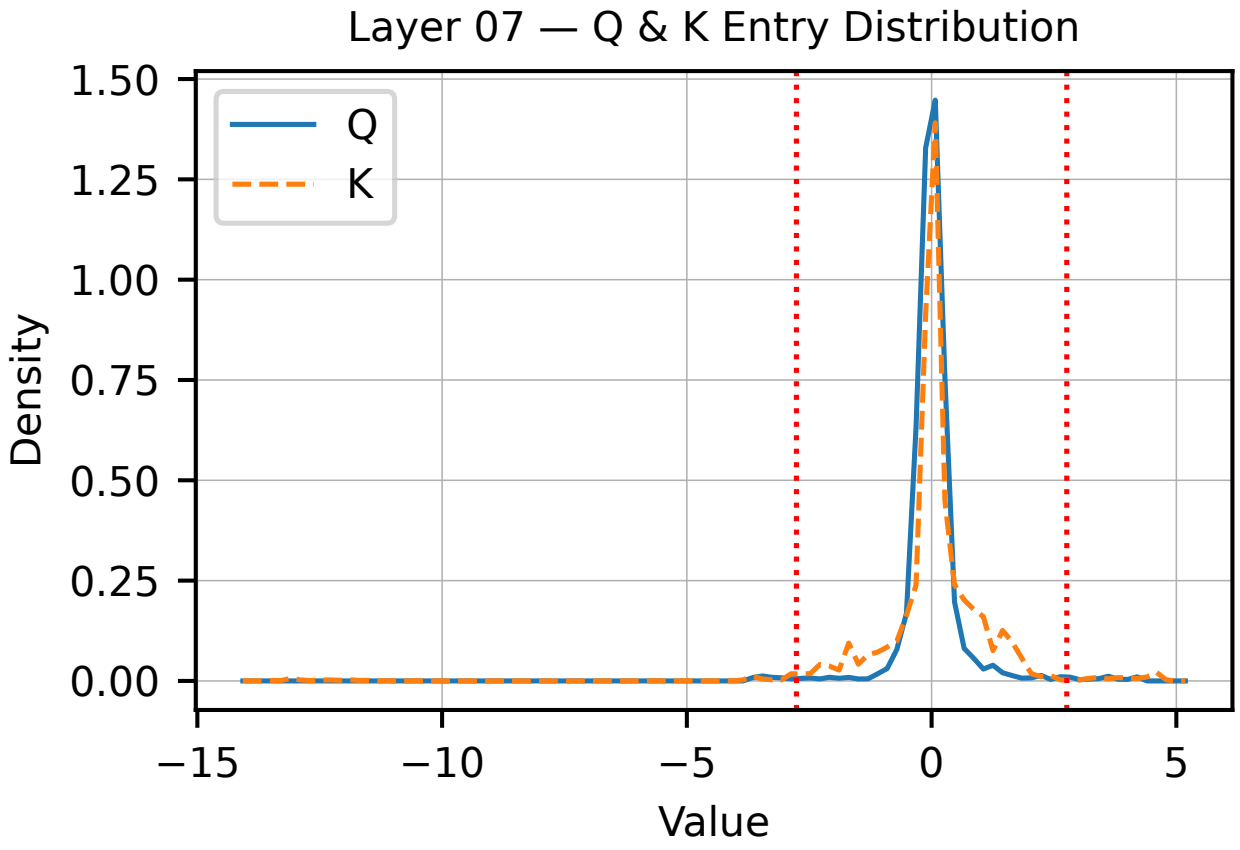


Figure 12: Distribution of entries in the query and key matrices of Layer 7 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

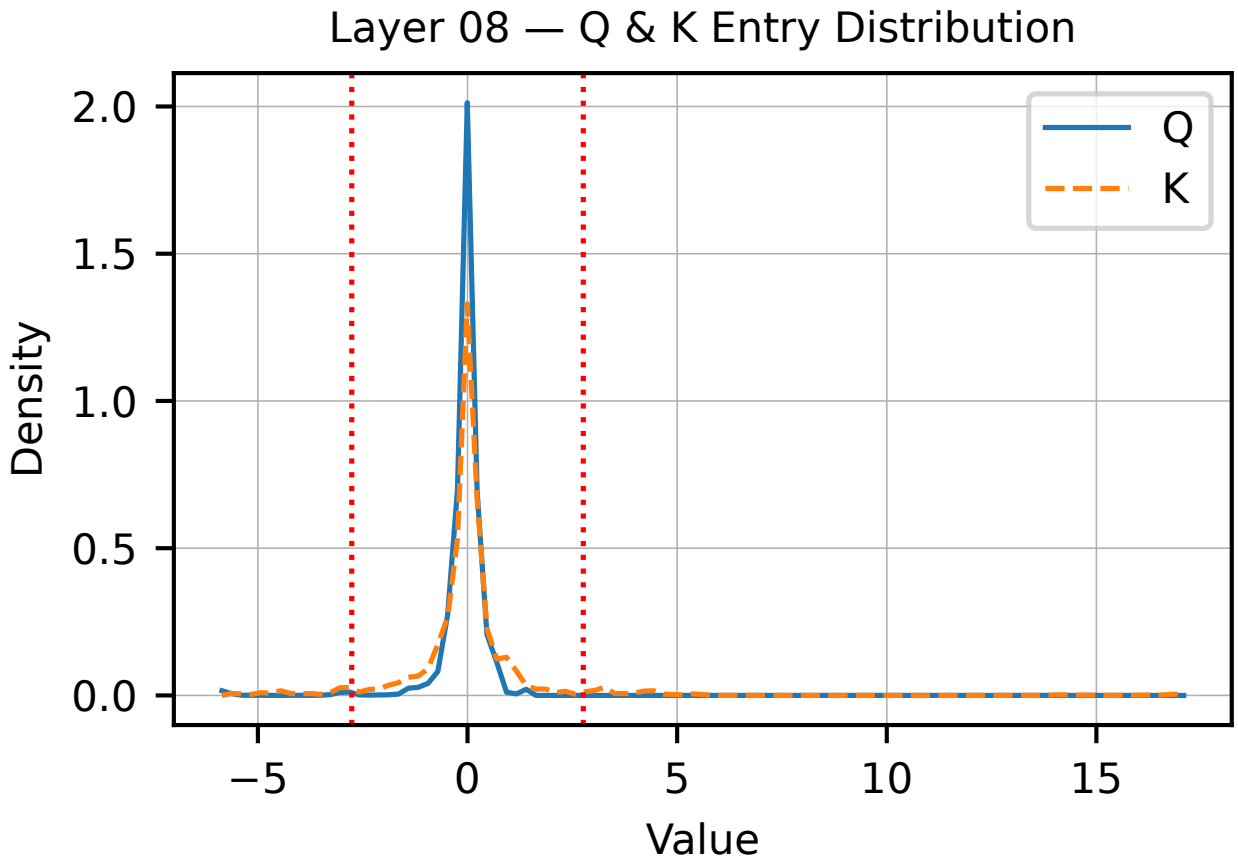


Figure 13: Distribution of entries in the query and key matrices of Layer 8 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

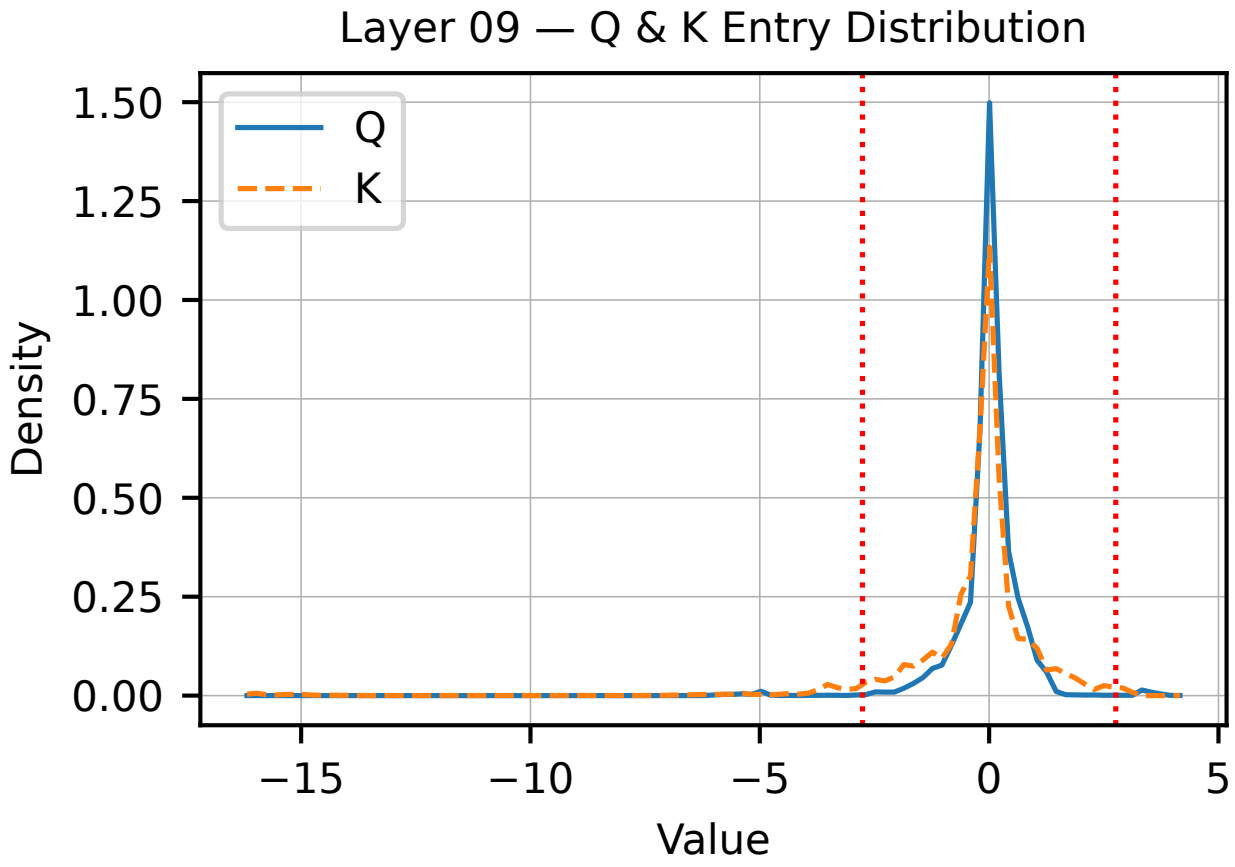


Figure 14: Distribution of entries in the query and key matrices of Layer 9 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

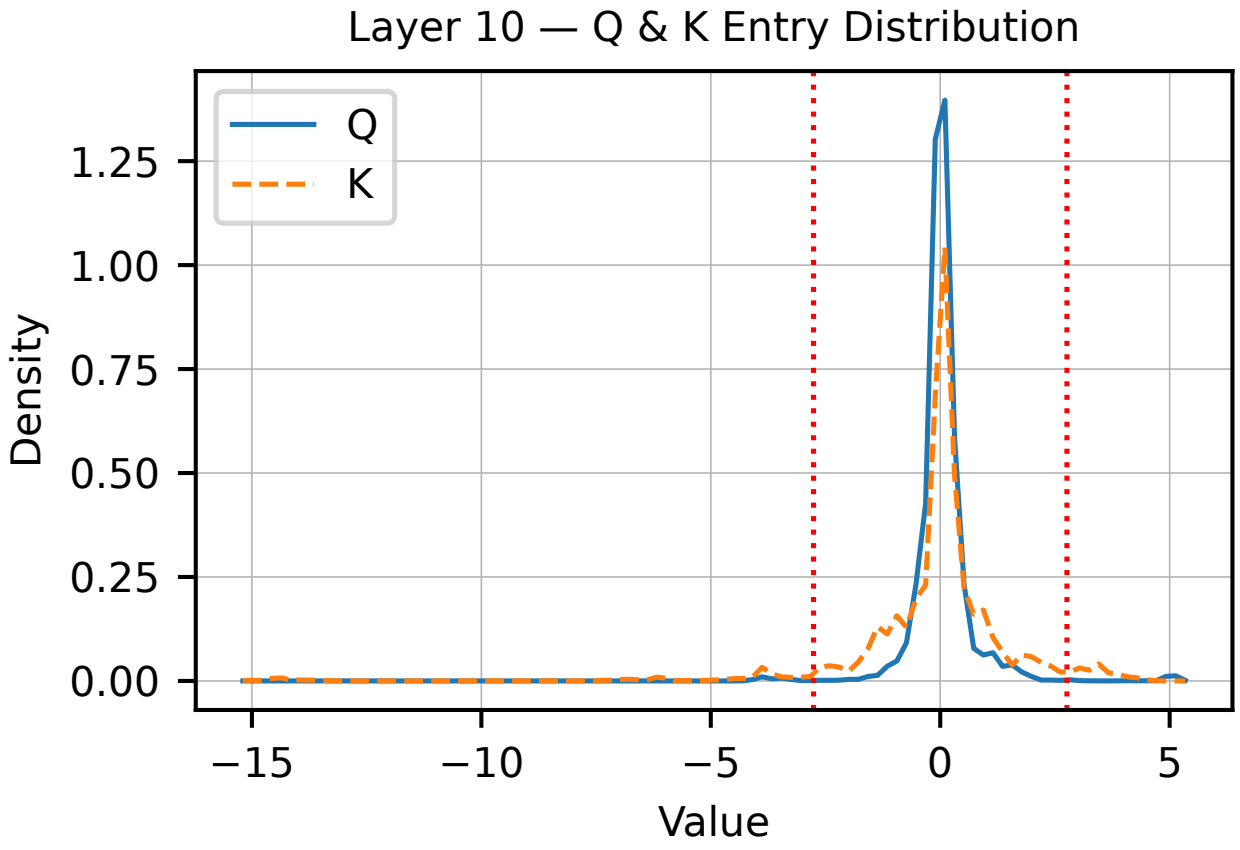


Figure 15: Distribution of entries in the query and key matrices of Layer 10 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

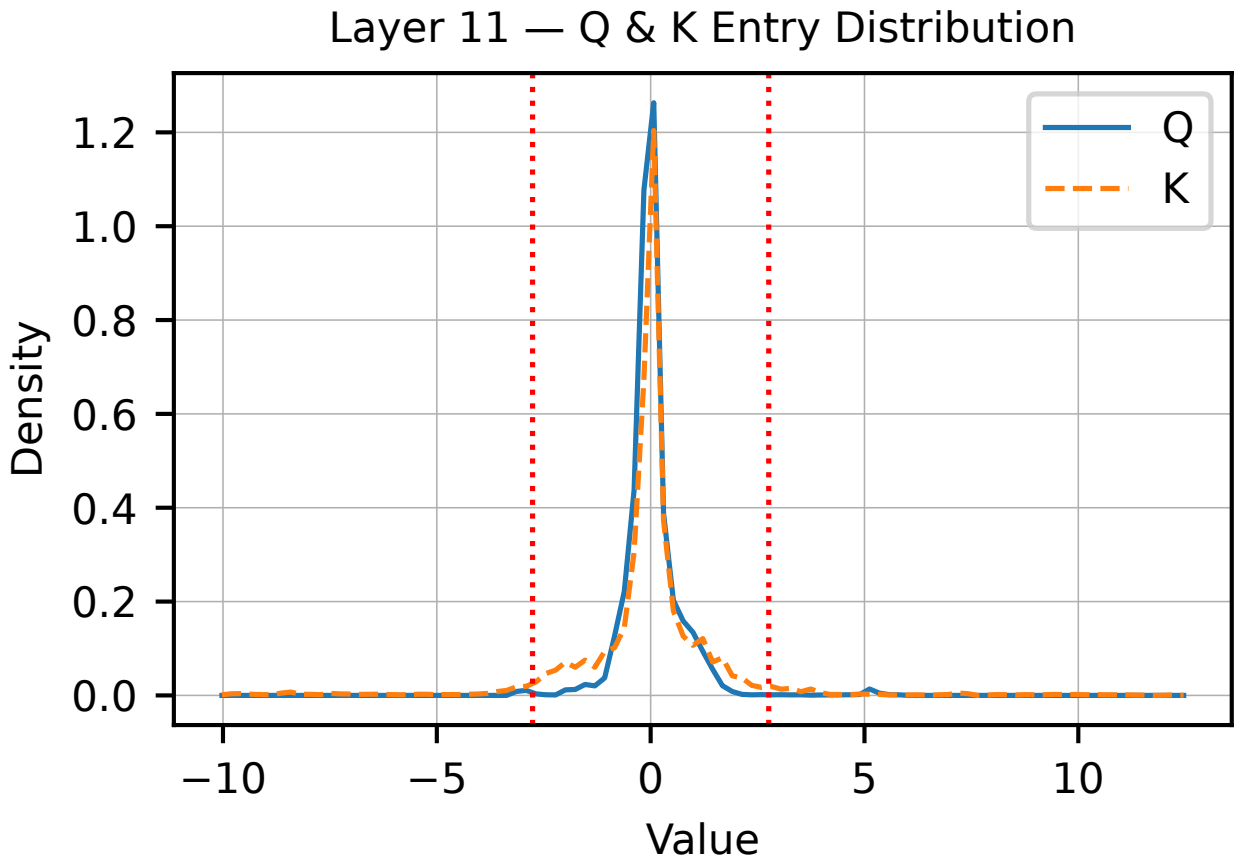


Figure 16: Distribution of entries in the query and key matrices of Layer 11 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

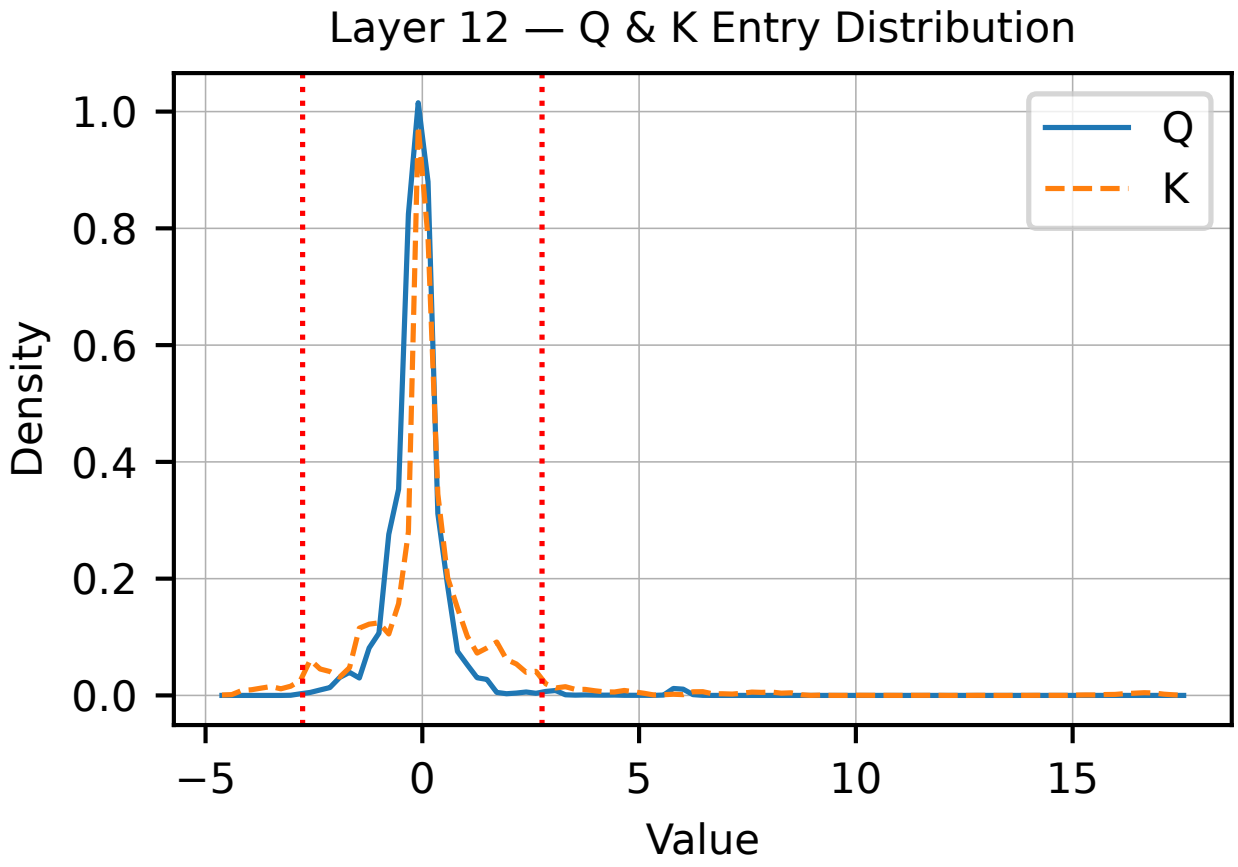


Figure 17: Distribution of entries in the query and key matrices of Layer 12 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

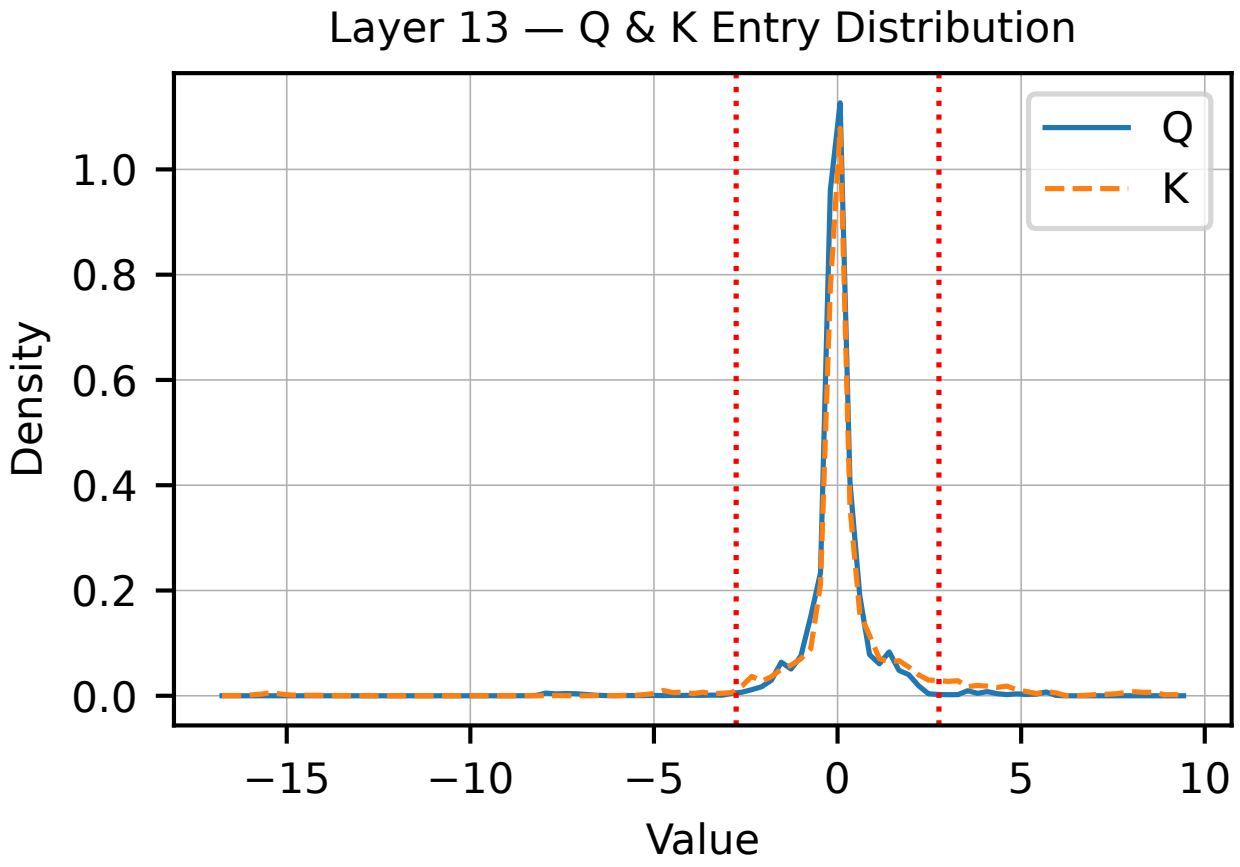


Figure 18: Distribution of entries in the query and key matrices of Layer 13 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

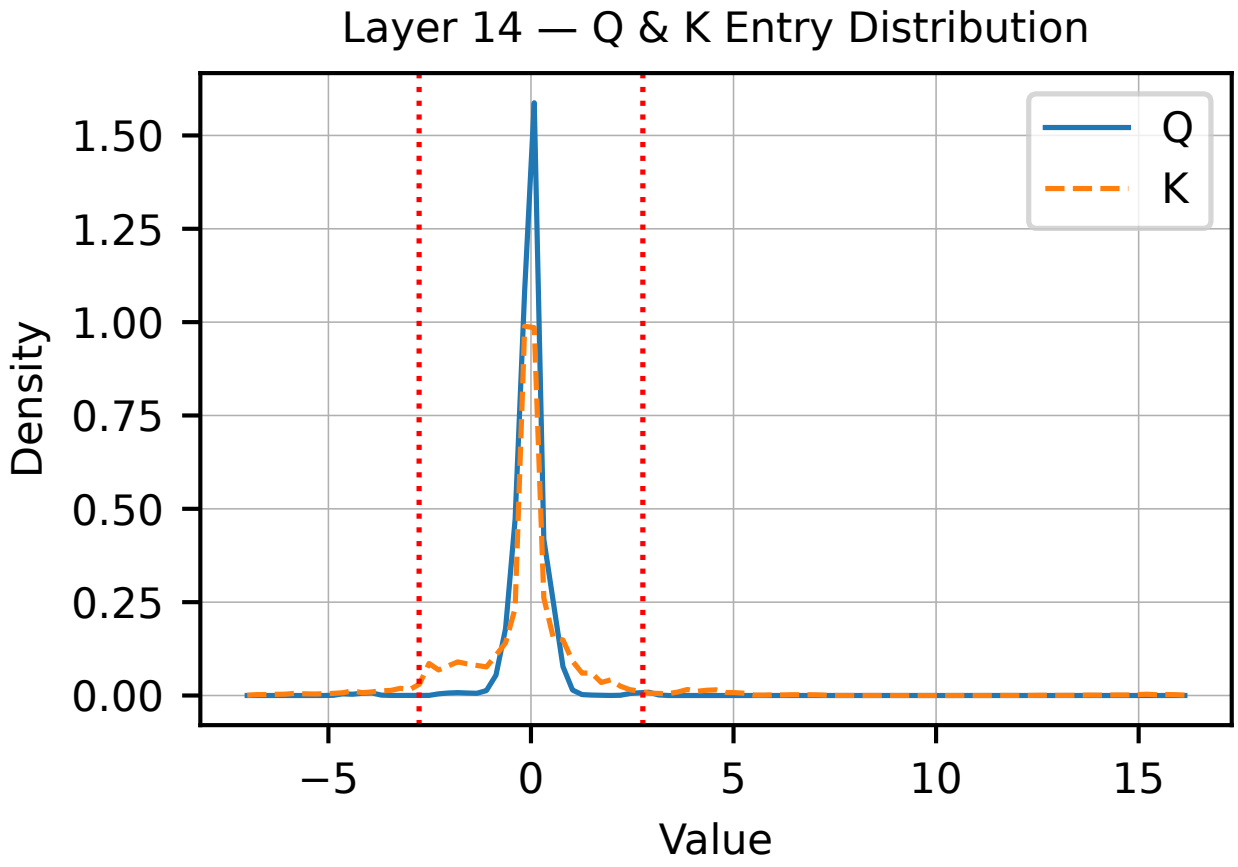


Figure 19: Distribution of entries in the query and key matrices of Layer 14 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

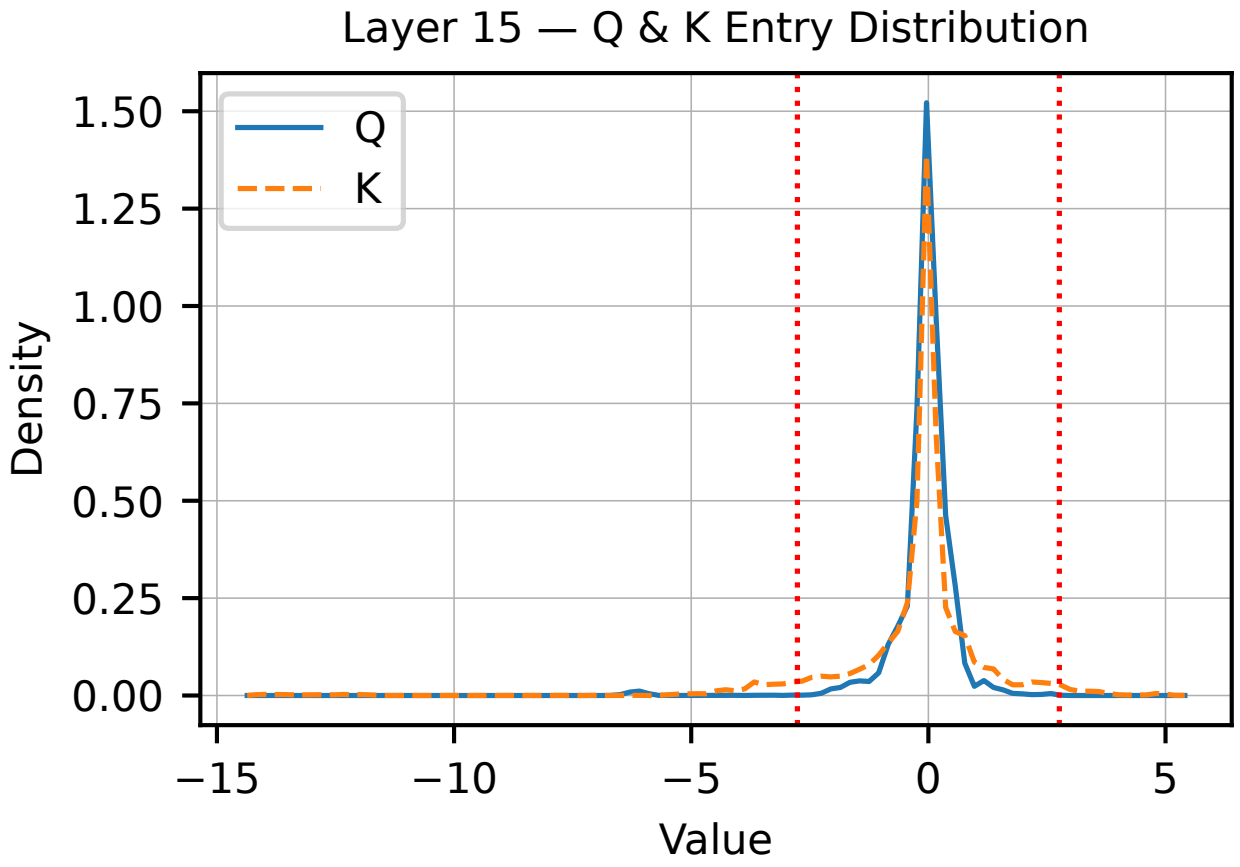


Figure 20: Distribution of entries in the query and key matrices of Layer 15 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

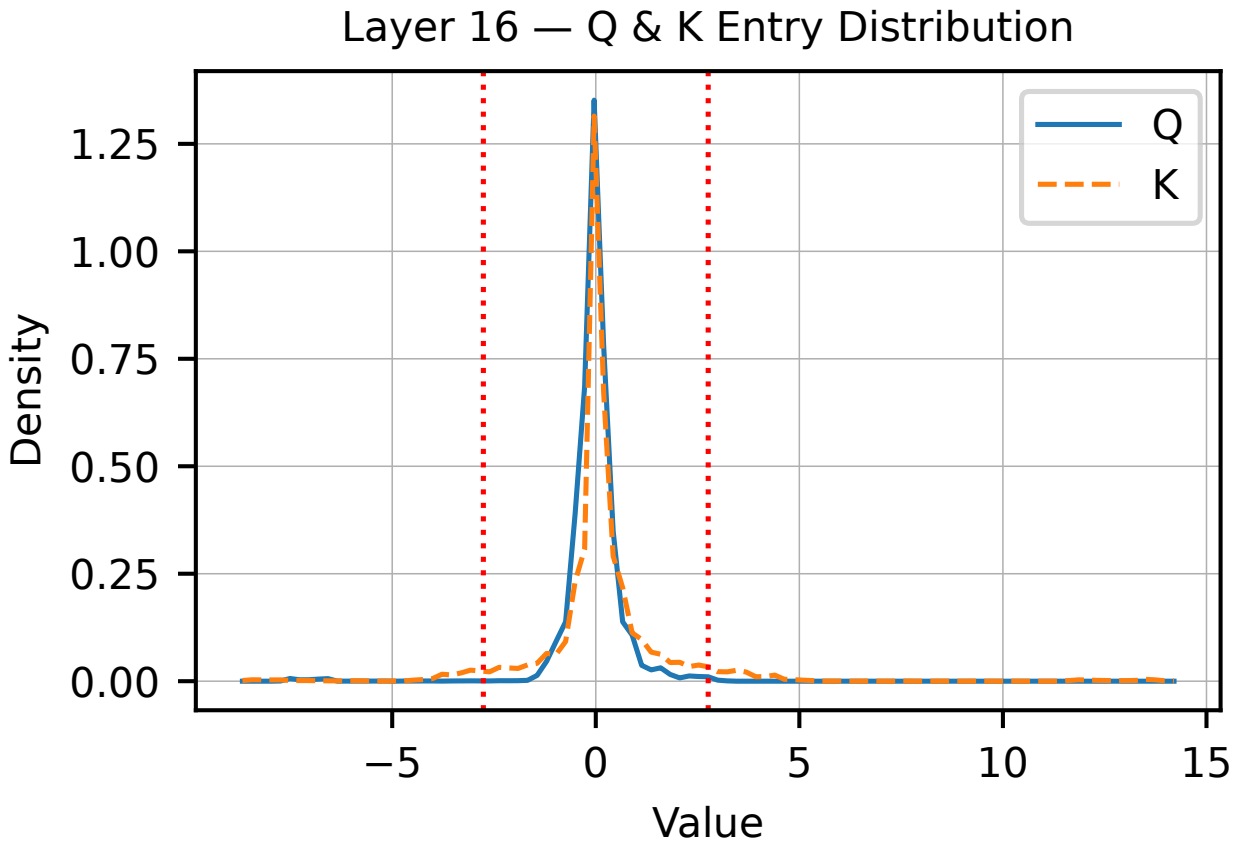


Figure 21: Distribution of entries in the query and key matrices of Layer 16 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

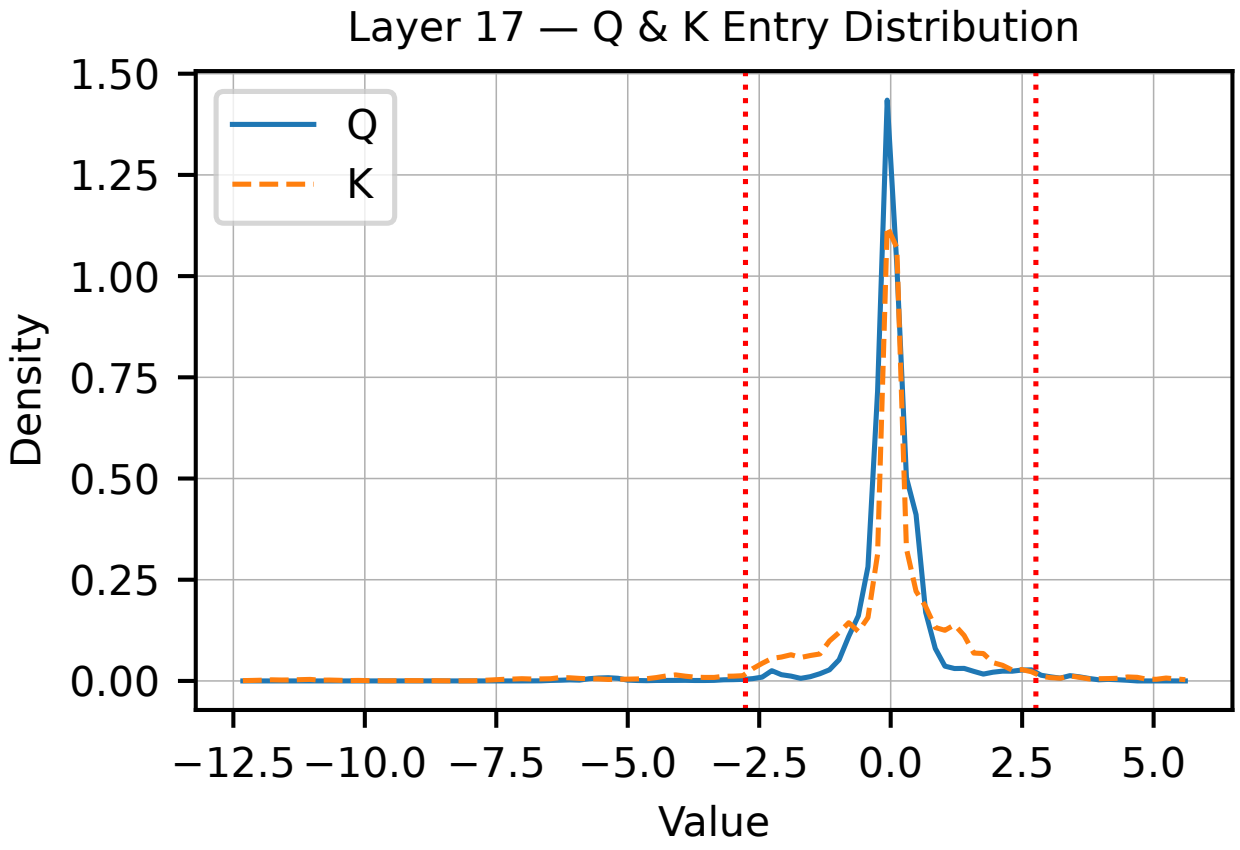


Figure 22: Distribution of entries in the query and key matrices of Layer 17 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

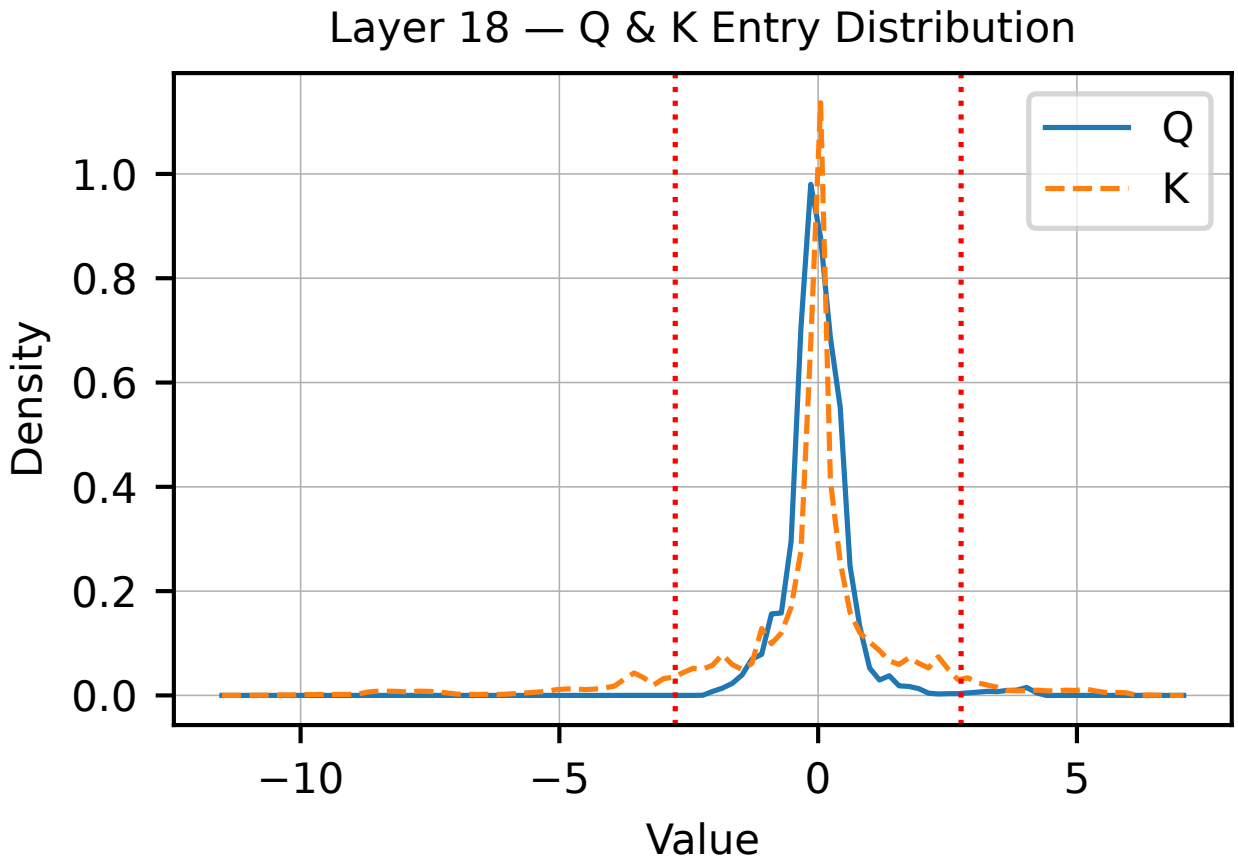


Figure 23: Distribution of entries in the query and key matrices of Layer 18 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

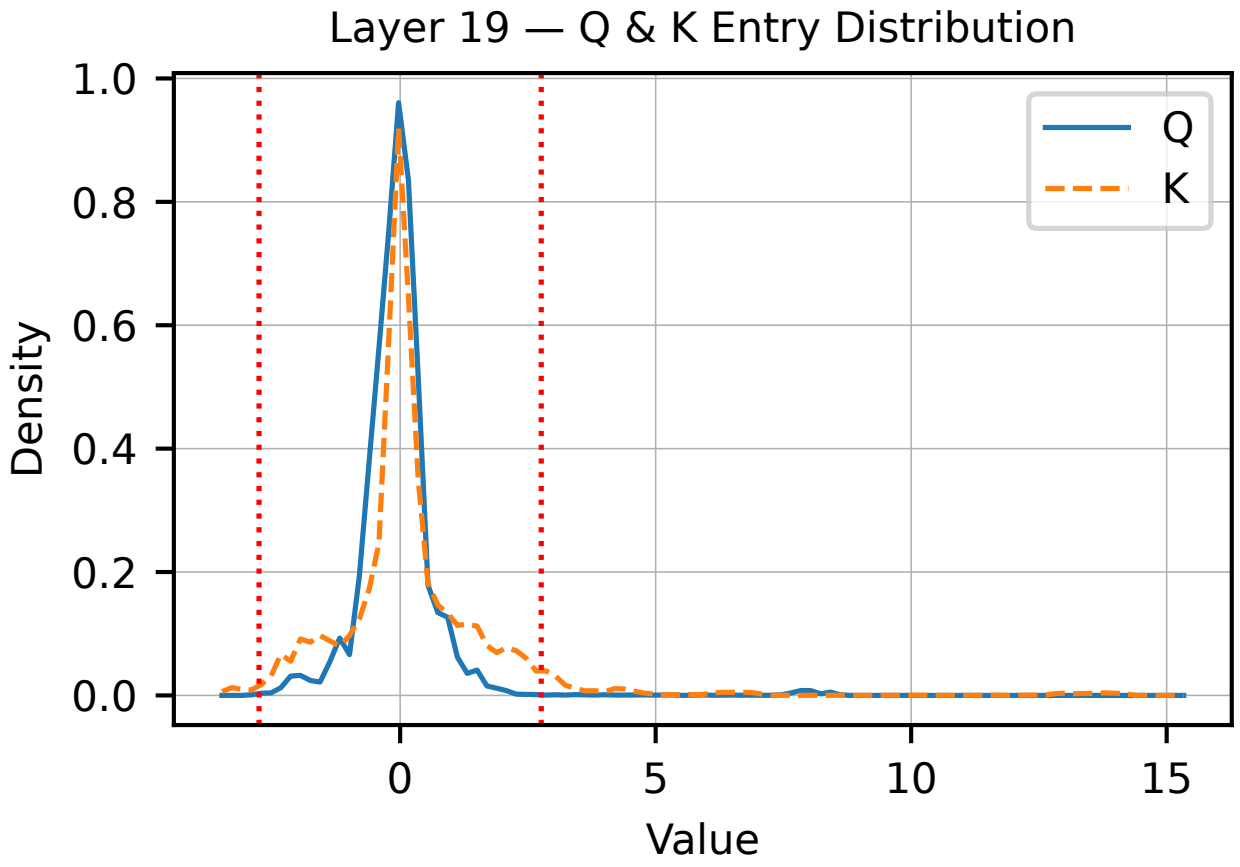


Figure 24: Distribution of entries in the query and key matrices of Layer 19 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

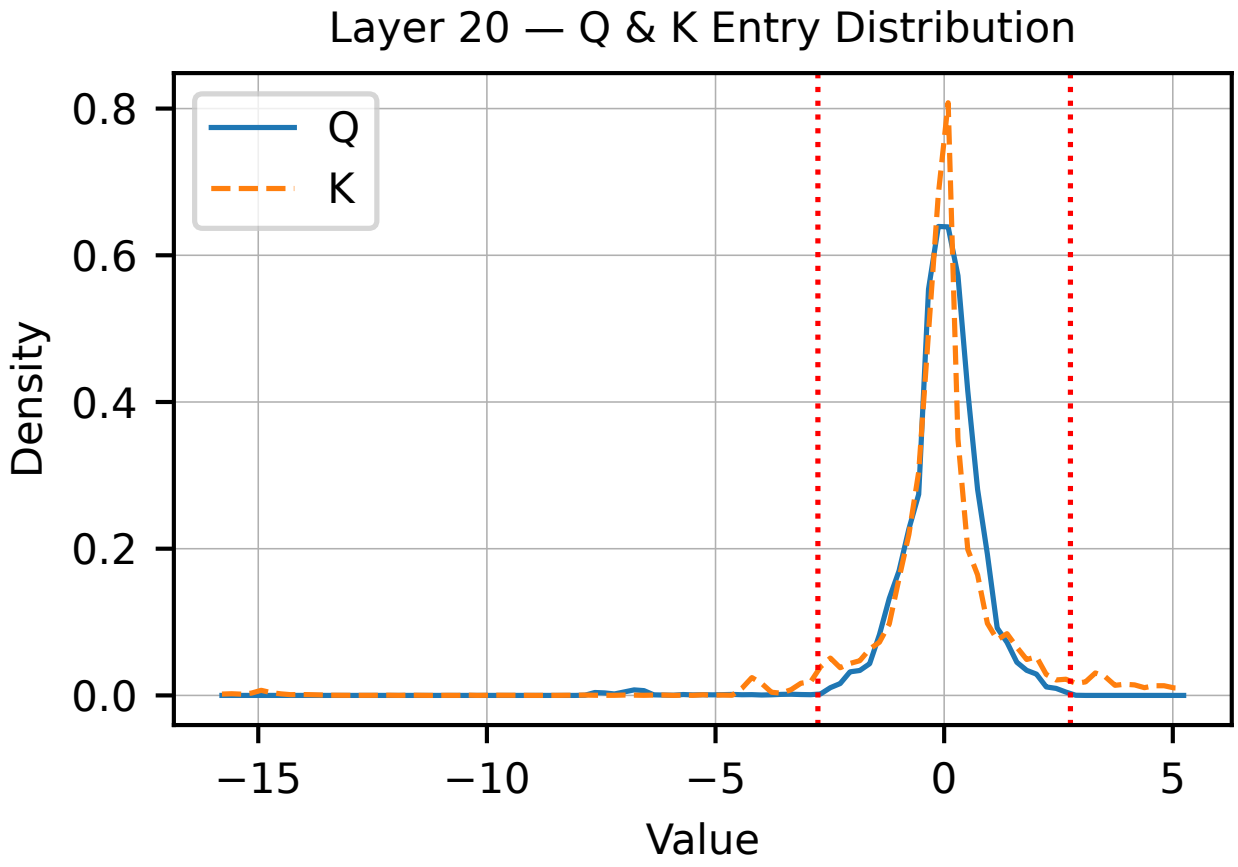


Figure 25: Distribution of entries in the query and key matrices of Layer 20 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

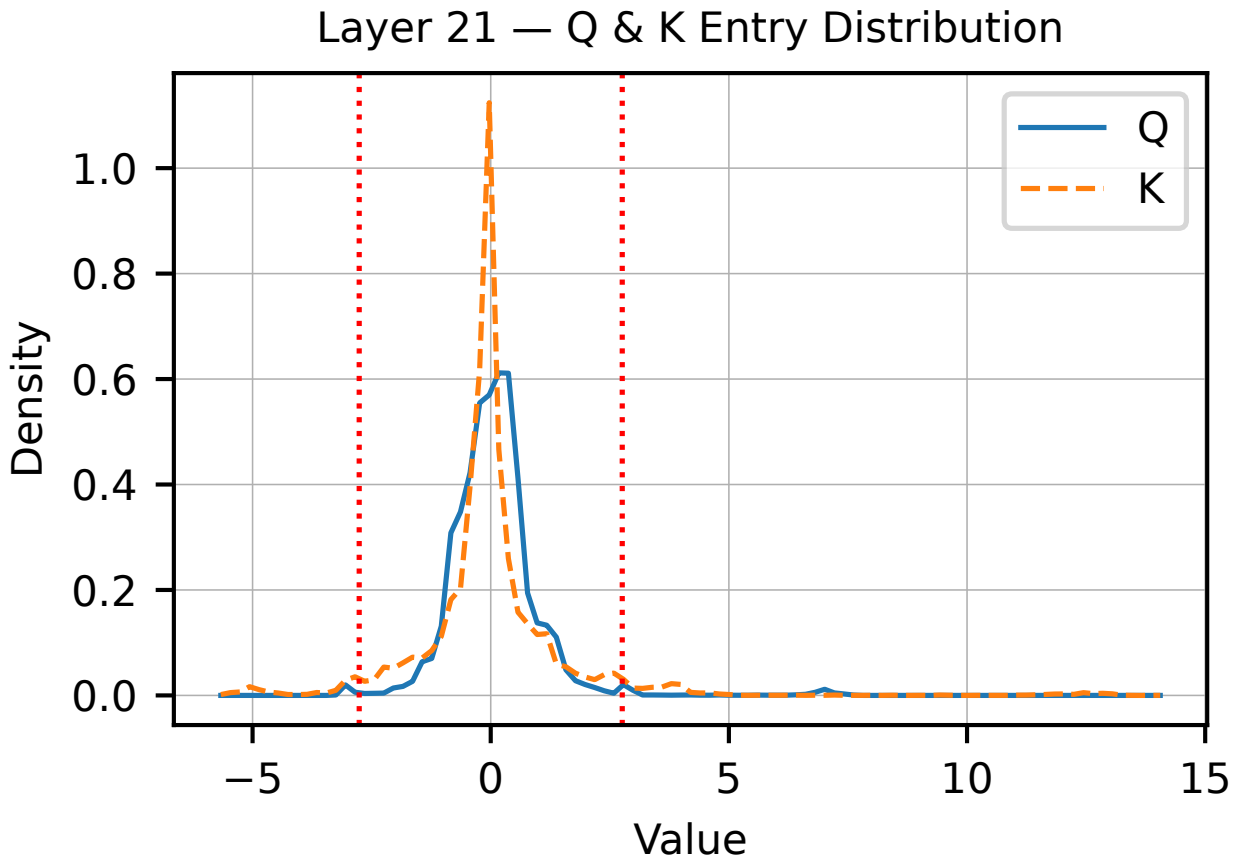


Figure 26: Distribution of entries in the query and key matrices of Layer 21 in TinyLlama-1.1B. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

L.2 TinyLlama-1.1B

In this section, we present the entry distribution of Q and K in TinyLlama-1.1B (Zhang et al., 2024) from layer 0 to layer 21. The token sequence consists of the following natural human-like tokens.

Natural human-like tokens

In the fast-paced world of modern research and technology, the ability to adapt quickly to new knowledge, integrate interdisciplinary ideas, and communicate them clearly has become a defining factor for success, not only for academic researchers but also for professionals working in industry, policy, or creative sectors. The challenge is no longer simply about having access to information—since the digital age has democratized knowledge to an unprecedented degree—but rather about cultivating the skills necessary to filter, evaluate, and synthesize the vast amount of data that is constantly flowing around us. A researcher today may begin the morning reading about advances in large language models, spend the afternoon designing experiments to validate theoretical insights, and finish the day by considering applications in medicine, finance, or education. This fluid movement between levels of abstraction demands both intellectual flexibility and a strong methodological foundation. At the same time, collaboration has emerged as a core feature of progress: breakthroughs increasingly arise not from the lone genius archetype, but from teams that combine different strengths, whether it is the theoretical rigor of mathematicians, the practical engineering sense of computer scientists, the domain knowledge of biologists, or the design intuition of human-computer interaction experts. Alongside collaboration, communication is equally essential. A brilliant idea poorly explained is often a wasted opportunity, while even a moderately novel insight presented with clarity and precision can influence the trajectory of a field. In this context, the role of writing, speaking, and visualizing cannot be underestimated. Writing a paper or report is not just a matter of documenting results but of shaping the interpretation and framing of those results, guiding how others will build upon them. Similarly, presenting at conferences, creating effective figures, and explaining technical ideas to broader audiences are all skills that expand the impact of one's work beyond immediate circles. Another layer to this landscape is the increasing pressure to balance depth with breadth. Specialization is necessary to push the frontier of a subfield, yet the most impactful research often arises from unexpected connections, such as applying methods from signal processing to neuroscience or adapting optimization techniques from physics to machine learning. To navigate this dual demand, researchers must cultivate a meta-skill: the ability to learn how to learn efficiently, to enter new fields without being overwhelmed, and to identify the key assumptions, tools, and open questions that define them. Underlying all of this is resilience and persistence, since research inevitably involves setbacks, failed experiments, and long periods of uncertainty. The process is rarely linear; rather, it is iterative and recursive, resembling more a spiral of refinement than a straight path toward discovery. In the end, success in modern research and professional life lies in the interplay of curiosity, rigor, creativity, and communication—qualities that allow individuals and teams not only to generate knowledge but to ensure that this knowledge becomes meaningful, usable, and transformative in the broader world.

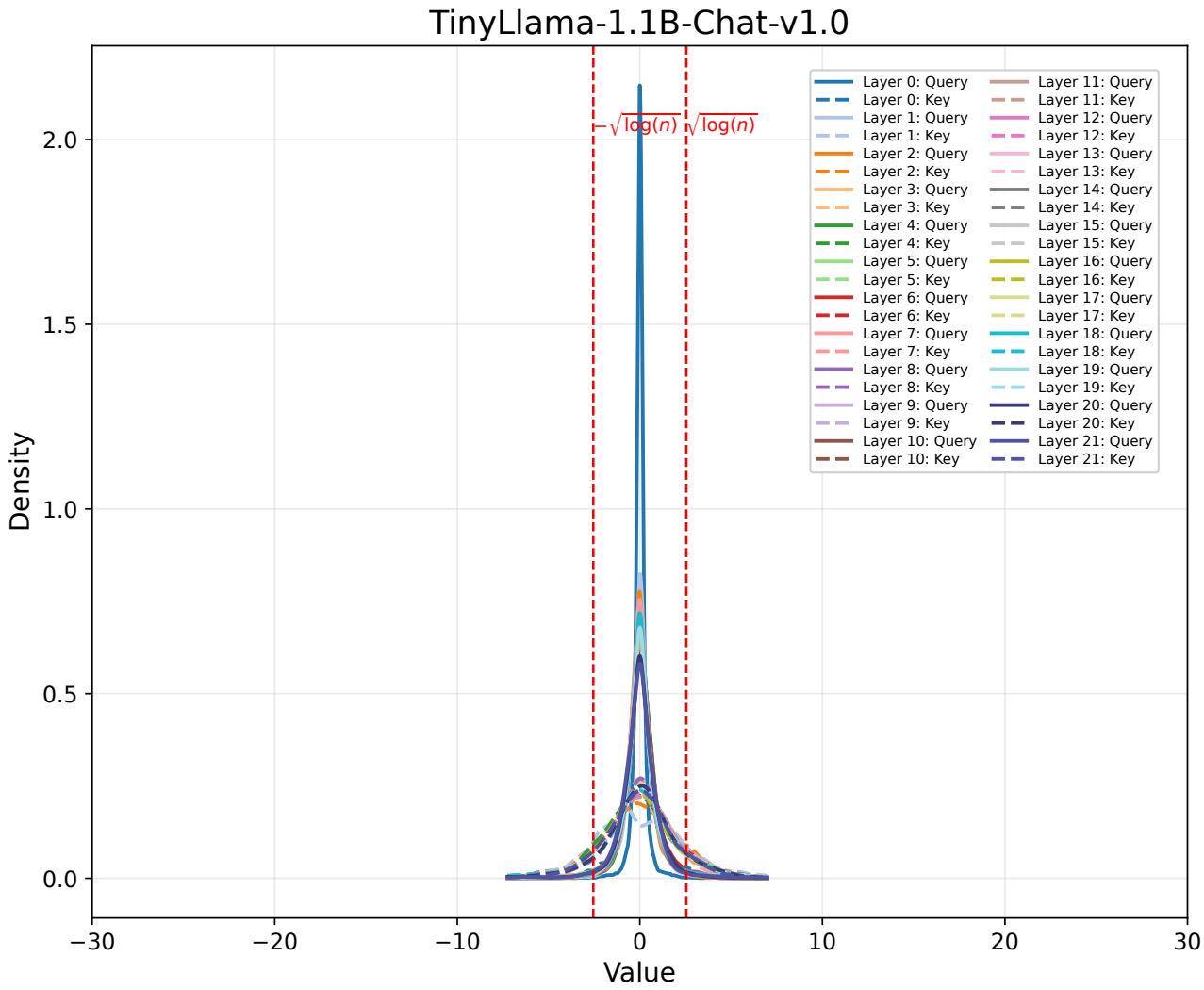


Figure 27: Distribution of entries in the query, key, and value matrices of TinyLlama-1.1B from layer 0 to layer 21. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

L.3 OPT-1.3B

In this section, we present the entry distribution of Q and K in OPT-1.3B (Zhang et al., 2022b) from layer 0 to layer 23. The token sequence consists of the natural human-like tokens.

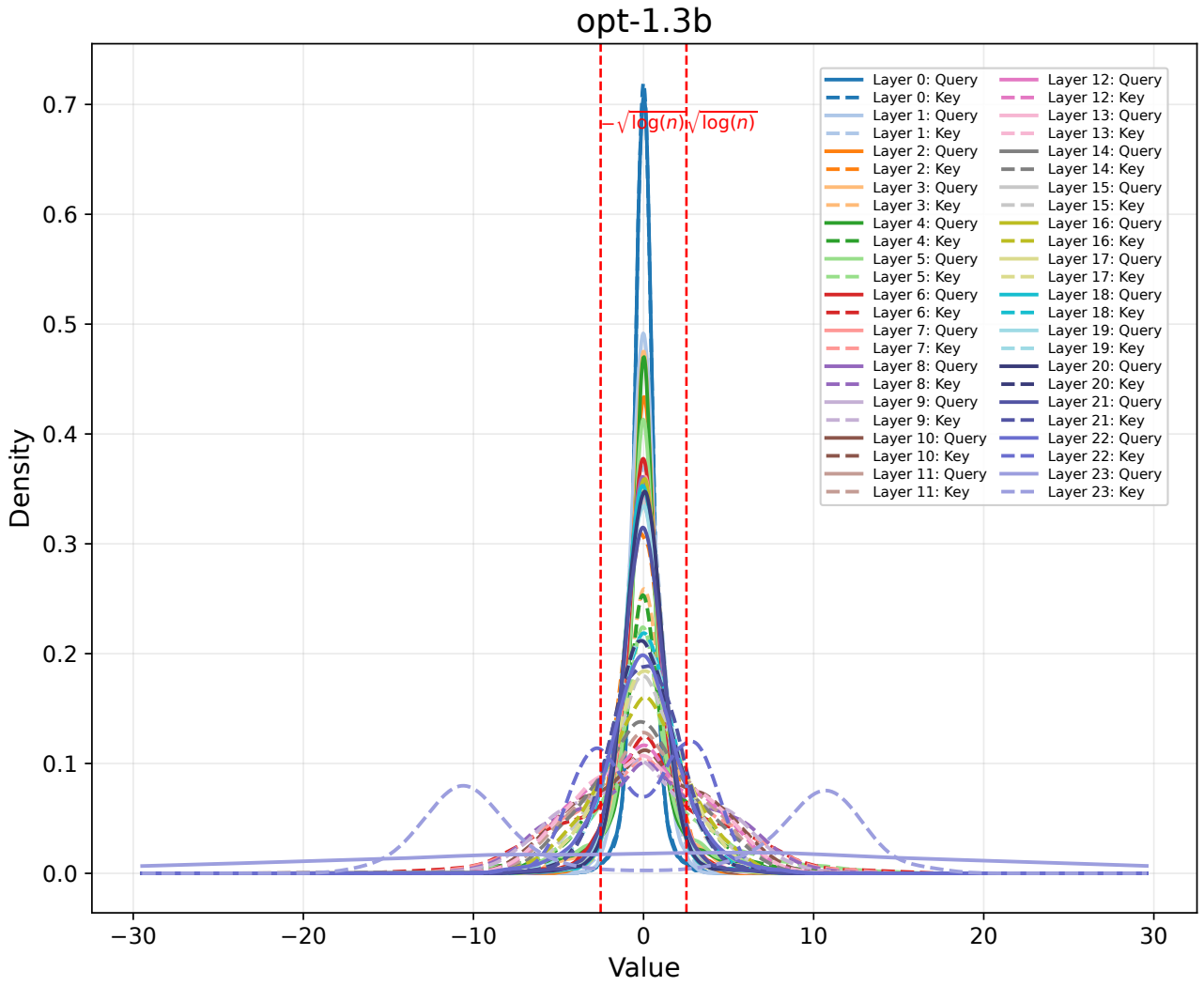


Figure 28: Distribution of entries in the query, key, and value matrices of OPT-1.3B from layer 0 to layer 23. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

L.4 LLaDA-8B-Base

In this section, we present the entry distribution of Q and K in LLaDA-8B-Base (Nie et al., 2025) from layer 0 to layer 31. The token sequence consists of the natural human-like tokens.

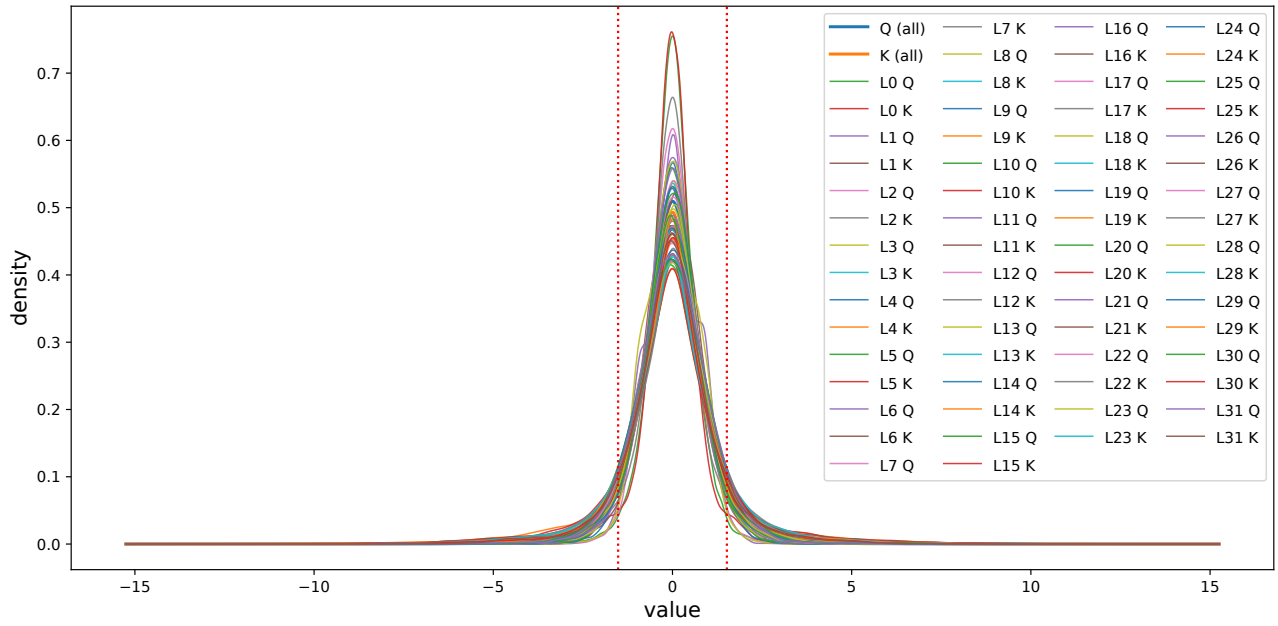


Figure 29: Distribution of entries in the query and key matrices of LLaDA-8B-Base from layer 0 to layer 31. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.

L.5 Phi-2

In this section, we present the entry distribution of Q and K in Phi-2 (Javaheripi et al., 2023) from layer 0 to layer 31. The token sequence consists of the natural human-like tokens.

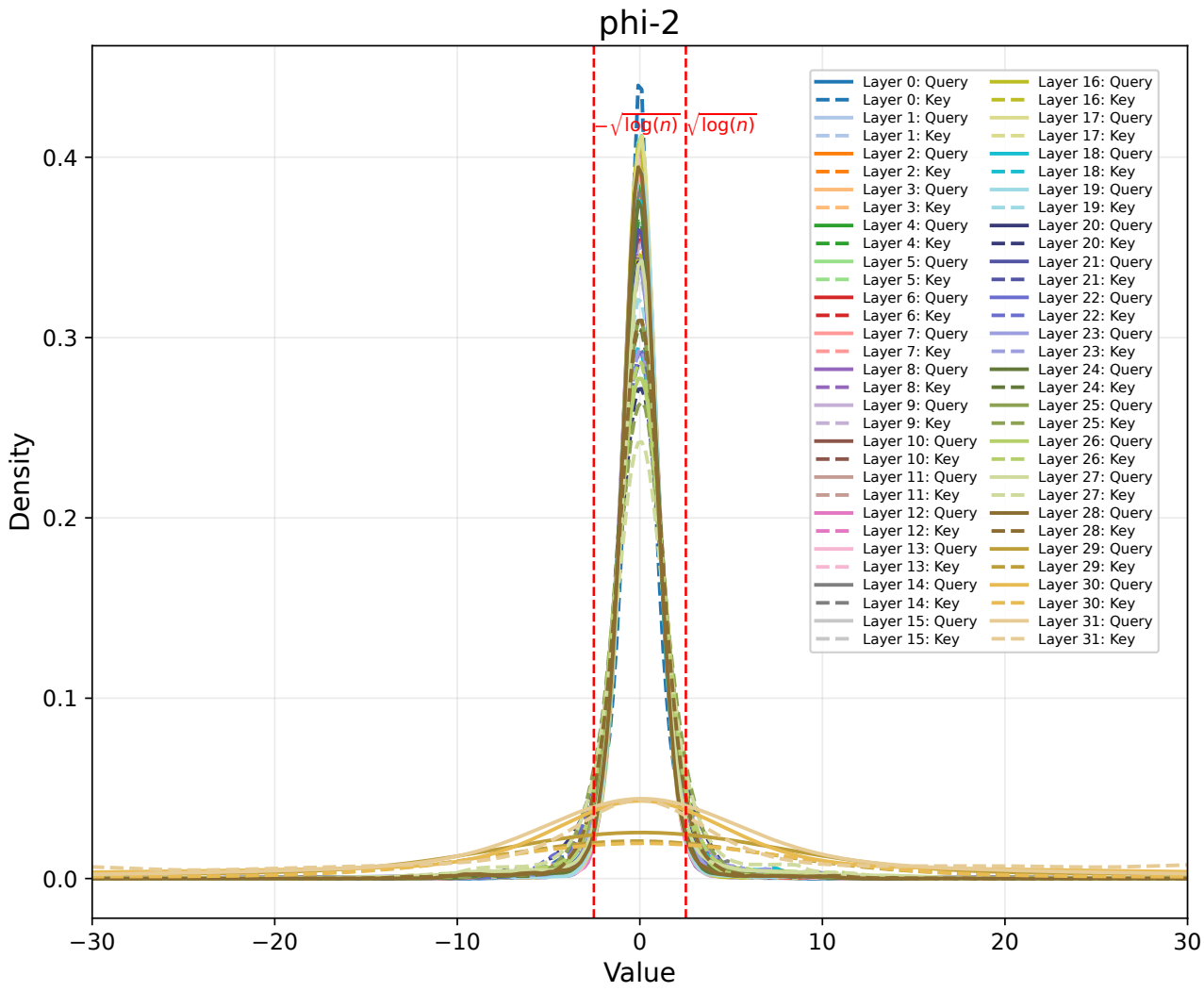


Figure 30: Distribution of entries in the query and key matrices of Phi-2 from layer 0 to layer 31. The red dashed lines mark the thresholds $\pm\sqrt{\log n}$.