EFFICIENTLY APPROXIMATING ATTENTION IS HARD

Anonymous authors
Paper under double-blind review

ABSTRACT

Softmax attention is ubiquitous in modern machine learning, but its quadratic scaling with sequence length makes it costly. To reduce this cost, attention is often approximated with fast algorithms, which incur error but can still perform well in practice and on *some* inputs. At the same time, the growing diversity of attention applications makes approximation guarantees that do not depend on particular input structure a compelling target. For such uniform guarantees over *all* inputs, known runtime lower bounds rule out fast algorithms for near-exact attention, but leave open the practically important regime: *is there an efficient algorithm with even a modest uniform approximation guarantee?* We answer this question negatively. Under standard complexity-theoretic assumptions, no truly subquadratic algorithm can approximate attention with any nontrivial additive or relative guarantee uniformly over all inputs. This impossibility holds in the mildest parameter regime for which known algorithms do not already achieve strong approximation guarantees in near-linear time, and extends to practically relevant relaxations: even after polynomial preprocessing of the KV cache, no efficient algorithm can obtain a nontrivial uniform approximation guarantee, or identify a small set of keys receiving substantial attention under sparsity. Overall, our results settle the computational limits of uniform attention approximation.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

The Transformer architecture has become the de facto standard for the most challenging sequence modeling problems across various domains (Vaswani et al., 2017; Devlin et al., 2019; Dosovitskiy et al., 2020; Gulati et al., 2020; Rives et al., 2021; Zhou et al., 2021), owing largely to its ability to model complex relationships between different points in a sequence. The key architectural component enabling this is *softmax attention*, which transforms a sequence of vectors based on pairwise comparisons. While attention makes Transformers powerful in practice, its standard implementation scales quadratically with the sequence length, making it the principal bottleneck of Transformer inference and often prohibitively expensive on long sequences (Beltagy et al., 2020; Dao, 2024).

To address this problem, various methods aim to approximate attention with fast algorithms, which can work well empirically on typical inputs (Kitaev et al., 2020; Xiong et al., 2021; Tay et al., 2020; Roy et al., 2021) or admit formal guarantees under structural assumptions on the inputs (Vyas et al., 2020; Chen et al., 2021; Han et al., 2024). In practice, however, Transformers are deployed across increasingly diverse applications and input distributions, making it difficult to test approximation quality exhaustively or to verify that data-dependent assumptions are satisfied. That makes algorithms particularly attractive if they can provide approximation guarantees *uniformly over all inputs*.

Remarkably strong approximation is already possible with uniform guarantees over sufficiently low-dimensional inputs with sufficiently small norms. In this regime, existing algorithms run in near-linear time, and guarantee error vanishing at polynomial or even super-polynomial rates in the sequence length (Alman & Song, 2023; Schröder & Mackey, 2026). Outside this regime, however, known lower bounds rule out such strong guarantees. More precisely, the best known lower bounds show that truly subquadratic algorithms, for some sequences of length n, must incur error n^{-C}, where $C \gg 0$ is a large constant (Alman & Song, 2023). While these results rule out extreme accuracy with fast algorithms, they leave a substantial gap at the approximation accuracy most relevant in practice. In particular, existing lower bounds do not even rule out a linear-time algorithm that uniformly guarantees error at most, say, n^{-100}. For all practical purposes, such high accuracy is effectively indistinguishable from exact computation, while empirical evidence suggests that Transformers remain robust even under much cruder approximations (Liu et al., 2024; Lin et al., 2022; Michel et al., 2019). This raises the question whether attention can be approximated efficiently once one tolerates a modest amount of error. We therefore ask:

Can attention be approximated efficiently with even a coarse uniform error guarantee?

We show that the answer is ***no: such an algorithm does not exist***. Concretely, we strengthen the proof of Alman & Song (2023) to show that, under standard complexity-theoretic assumptions, no algorithm running in truly subquadratic[1] time can give *any* nontrivial approximation guarantee uniformly over all inputs. Our result holds for both additive and relative error notions. Moreover, it applies under minimal assumptions on the data dimension and norms: even the slightest relaxation of these assumptions enters the setting in which existing algorithms supply near-linear runtime and vanishingly small error. Consequently, our new lower bound establishes a sharp transition of uniform attention approximation from possible at high accuracy to impossible at any accuracy.

Additionally, we study whether preprocessing or sparse attention structure can accelerate subsequent attention computations with uniform guarantees. Such approaches are common in practice, where one is often willing to incur the quadratic cost of initially processing the input, as well as additional preprocessing overhead, in exchange for faster inference on subsequent queries (Li et al., 2024; Tang et al., 2024; Chen et al., 2025; Liu et al., 2025). We show that, even in this relaxed setting, uniform approximation remains hard. Under the same mild assumptions as before, it is impossible to preprocess the input in polynomial time and subsequently approximate attention in truly sublinear time per token while guaranteeing either a nontrivial approximation of the attention output or, under significant sparsity, retrieval of a small set of keys capturing a constant fraction of the attention.

Contributions. We provide new lower bounds that settle the limits of uniform attention approximation, closing the gap left by prior work and extending them to autoregressive decoding with preprocessing and sparsity. Beyond the regimes where uniform high-precision approximation is already efficiently possible, we show that approximate attention must exploit data-dependent structure.

[1] By "truly" subquadratic, we mean polynomially faster, that is, $\mathcal{O}(n^{2-q})$ for some constant $q > 0$.

2 RELATED WORK

Fast approximate attention. The quadratic cost of the standard attention implementation has motivated extensive work on faster approximations. For instance, existing approaches exploit sparsity (Child et al., 2019; Zaheer et al., 2020; Tay et al., 2020; Kitaev et al., 2020; Roy et al., 2021), low-rank structure (Wang et al., 2020; Xiong et al., 2021; Chen et al., 2021), or other data-dependent properties of the attention matrix (Zandieh et al., 2023; Han et al., 2024; Aliakbarpour et al., 2026) for efficient attention approximation. These methods can substantially reduce computation while preserving model quality, but they demonstrate performance often only empirically for typical inputs or give guarantees that depend on structural properties of the input. In particular, most approaches do not provide uniform guarantees over all inputs even though such guarantees are appealing given the use of attention across domains as varied as language, vision, speech, biology, and time-series forecasting (Devlin et al., 2019; Dosovitskiy et al., 2020; Gulati et al., 2020; Rives et al., 2021; Zhou et al., 2021). This raises **Question (A):** *is the dependence on input structure merely a limitation of existing algorithms or fundamentally unavoidable in practice?*

Uniform approximation guarantees. Another line of work gives approximation guarantees uniformly over a broad class of inputs, typically satisfying a fixed norm bound without requiring additional instance-specific structure. For instance, Choromanski et al. (2021) prove uniform convergence of a random-feature approximation when the query and key norms are bounded, although the number of features depends strongly on the radius and desired accuracy. Alman & Song (2023) obtain a near-linear time algorithm with approximation error decaying at a rate of n^{-c} in the sequence length n for a configurable constant $c > 0$ when the entries of the key and query vectors have sufficiently small magnitude. More recently, Schröder & Mackey (2026) achieve even super-polynomial error decay in the sequence length when keys and queries have sufficiently small norms. While these algorithms give approximation guarantees uniformly over all inputs, they have in common that they only result in fast algorithms for sufficiently low-dimensional inputs with sufficiently small norms. This raises **Question (B):** *are these restricted parameter regimes merely a limitation of existing algorithms or are they the broadest possible for uniform approximation guarantees?*

Runtime lower bounds. Existing runtime lower bounds provide evidence that approximating attention is hard, but do not conclusively answer Questions **(A)** or **(B)**. Early work by Duman Keles et al. (2023) establishes that near-exact attention computation requires near-quadratic runtime when the input dimension and norms are sufficiently large. More directly, Alman & Song (2023) show that, outside the parameter regime in which their algorithm guarantees arbitrary polynomial error decay, there exists some constant $C \gg 0$ such that no truly subquadratic algorithm can approximate attention with error at most n^{-C} uniformly over all input sequences of length n. These results concern high-precision approximation, since Alman & Song (2023) quantify C only existentially, while Duman Keles et al. (2023) only rule out fast algorithms with even faster-decaying additive error. More recent lower bounds by Gupta et al. (2026, Thm. 5.4) rule out constant-error approximation, but only for inputs with extremely large norms, while their complementary results in more realistic regimes (Gupta et al., 2026, Thm. 5.7) again require extreme precision. In contrast, our new lower bound rules out *any* nontrivial additive or relative error guarantee, thereby resolving Questions **(A)** and **(B)**. At a technical level, our results strengthen the proof of Alman & Song (2023) and therefore build on the lower bounds of Rubinstein (2018) for the approximate closest pair problem.

Approximation with preprocessing and sparsity. In practice, the runtime of Transformers on long sequences is often not dominated by the processing of the initial input but instead by the cost of subsequent queries during generation, since the former can be parallelized across the sequence whereas the latter cannot. Because of that, significant effort has gone into designing algorithms that speed up generation, often by preprocessing the sequence into some data structure that subsequently exploits sparsity to attend only over a small subset of the original input (Li et al., 2024; Tang et al., 2024; Chen et al., 2025; Liu et al., 2025). These methods incur the full quadratic cost of attention before generation, as well as the added cost of preprocessing, in exchange for faster generation. Prior lower bounds leave open whether relaxing the computational problem in this way enables fast attention approximation with uniform guarantees during generation. We address this gap by extending our runtime lower bounds to the relaxed problem settings of approximate attention with preprocessing and retrieval under sparsity.

3 NONTRIVIAL GUARANTEES ARE IMPOSSIBLE UNDER MILD ASSUMPTIONS

In this section, we ask whether a fast algorithm can approximate attention uniformly over all inputs. After formally defining the problem in Section 3.1, we explain in Section 3.2 why existing lower bounds do not resolve this question. Our main result in Section 3.3 then strengthens the lower-bound argument of Alman & Song (2023) to rule out any nontrivial additive or relative approximation guarantee in truly subquadratic time.

3.1 PROBLEM FORMULATION

We consider the standard softmax attention operation over a sequence of input vectors. Attention allows each vector to selectively aggregate information from all other vectors in the sequence. To determine what information to aggregate, each element exposes a *query*, a *key*, and a *value*. Each query is compared with all keys, and the resulting scores determine how the corresponding values are combined. Formally, this operation is defined as follows.

Definition 3.1 (Attention)**.** *For matrices $\boldsymbol{Q} \in \mathbb{R}^{m\times d}$ and $\boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{n\times d}$, let $\boldsymbol{A} = \exp(\boldsymbol{Q}\boldsymbol{K}^\top/d)$ and $\boldsymbol{D} = \operatorname{diag}(\boldsymbol{A}\mathbf{1}_n)$, where $\exp$ is applied entrywise. Define* attention *as*

$$\operatorname{Att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) \coloneqq \boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{V} \in \mathbb{R}^{m\times d}.$$

Remark 3.2. *For better comparability of our results with prior work, we follow Alman & Song (2023, Def. 1.1) in normalizing $\boldsymbol{Q}\boldsymbol{K}^\top$ by d. In practice, normalization often uses $\sqrt{d}$ or a tunable parameter, instead. The precise normalization is inconsequential for our results, however, because it can always be absorbed into the keys and queries by suitable scaling.*

The straightforward implementation of attention computes the full attention matrix $\boldsymbol{A}$, comparing each of the m queries to each of the n keys before aggregating the values. In the usual setting where $m = n$, this results in $\mathcal{O}(n^2)$ operations. At the same time, both the input and output only have $\mathcal{O}(nd)$ entries and the embedding dimension d is often much smaller than the sequence length n. Consequently, one may hope for an algorithm that avoids explicitly forming all n^2 comparisons and computes $\operatorname{Att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})$, at least approximately, in much less than $\mathcal{O}(n^2)$ time. We formalize this problem below for different uniform approximation guarantees such an algorithm could give.

Problem 1: Approximating attention

Fix parameters $n, d \in \mathbb{N}$ and $B \geq 0$. Given matrices $\boldsymbol{Q}, \boldsymbol{K} \in [-B, B]^{n\times d}$ and $\boldsymbol{V} \in [0,1]^{n\times d}$, the task is to compute a matrix $\boldsymbol{T} \in \mathbb{R}^{n\times d}$. For this output, the problem

$\mathsf{AbsAtt}(n, d, B, \eta_{\mathrm{abs}})$	requires	$\lVert \boldsymbol{T} - \operatorname{Att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})\rVert_{\ell_\infty} \leq \eta_{\mathrm{abs}}$, and
$\mathsf{RelAtt}_p(n, d, B, \eta_{\mathrm{rel}})$	requires	$\lVert \boldsymbol{T} - \operatorname{Att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})\rVert_{\mathcal{S}_p} \leq \eta_{\mathrm{rel}} \lVert \operatorname{Att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})\rVert_{\mathcal{S}_p}$.

We allow an algorithm to approximate attention with either an additive or a relative error guarantee. For additive error, we follow prior work in measuring error entrywise in ℓ_∞ norm and restrict the values to $[0, 1]$, since otherwise the error could be inflated arbitrarily by scaling the values. For relative error, we instead allow any Schatten norm $\mathcal{S}_p$, covering a broad range of natural relative and spectral guarantees not addressed by previous lower bounds. Under these conventions, additive error $1/2$ and relative error 1 are trivial to achieve with no dependence on the input. A useful algorithm should therefore guarantee additive error below $1/2$ or relative error below 1.

Our goal is to show that an algorithm with such nontrivial approximation guarantees cannot run in truly less than $\mathcal{O}(n^2)$ time, under the mildest possible assumptions on d and B. Concretely, we target $d = \Theta(\log n)$ and $B = \Theta(\sqrt{\log n})$, since making either parameter asymptotically smaller yields known algorithms with near-linear runtime and with vanishingly small error (Alman & Song, 2023; Schröder & Mackey, 2026). To rule out efficient algorithms at this boundary, we work under the strong exponential time hypothesis (SETH), which is a standard assumption in fine-grained complexity theory. In particular, essentially all prior lower bounds for attention are conditional on SETH (Alman & Song, 2023; Duman Keles et al., 2023; Gupta et al., 2026).

Hypothesis 3.3 (SETH, Impagliazzo & Paturi (2001))**.** *For every $\varepsilon > 0$, there is an integer $k \geq 3$ such that k-SAT on formulas with n variables cannot be solved in $\mathcal{O}(2^{(1-\varepsilon)n})$ time, not even by a randomized algorithm with bounded error probability.*

Table 1: Comparison of hardness results for approximating attention. All results establish $n^{2-o(1)}$ runtime lower bounds under SETH, but apply to different settings (head dimensions d and entry bounds B) and rule out different error guarantees. For comparability, all parameters are standardized to $\exp(\boldsymbol{Q}\boldsymbol{K}^\top/d)$ normalization. "Error scale" reports the error threshold below which hardness is established. For the exact attention output $\boldsymbol{Y} \in \mathbb{R}^{n\times d}$ and an approximation $\boldsymbol{T} \in \mathbb{R}^{n\times d}$, additive error means $\|\boldsymbol{T}-\boldsymbol{Y}\|_{\ell_\infty}$; entrywise-relative error means the smallest η such that $|\boldsymbol{T}_{ij}-\boldsymbol{Y}_{ij}| \le \eta|\boldsymbol{Y}_{ij}|$ for all i, j; and norm-relative error means $\|\boldsymbol{T}-\boldsymbol{Y}\|_{\mathcal{S}_p}/\|\boldsymbol{Y}\|_{\mathcal{S}_p}$ for any Schatten norm $\mathcal{S}_p$. Additive errors are comparable because all constructions use value entries in $[0, 1]$. All previous results either apply only under extreme temperature or rule out only extremely high-precision approximation. We establish hardness in the mildest setting and for the coarsest error scales, ruling out *every nontrivial approximation guarantee* under all three error notions.

	Setting		**Error scale**		
Result	d	B	Additive	Entrywise relative	Norm relative
Gupta et al. (2026) Thm. 5.4	$2^{\Theta(\log^* n)}$	$n^{\Theta(1)}$	$\Theta(1)$	$\Theta(1)$	–
Gupta et al. (2026) Thm. 5.7	$\Theta(\log n)$	$\Theta(\sqrt{\log n})$	$n^{-\Theta(1)}$	$n^{-\Theta(1)}$	–
Duman Keles et al. (2023)	$\omega(\log n)$	$\Theta(d^{3/2})$	$n^{-\Omega(d)}$	1	–
Alman & Song (2023)	$\Theta(\log n)$	$\Theta(\sqrt{\log n})$	$n^{-\Theta(1)}$	$n^{-\Theta(1)}$	–
Theorem 3.7 (ours)	$\Theta(\log n)$	$\Theta(\sqrt{\log n})$	$\frac{1}{2}$	1	1

3.2 Existing lower bounds only rule out high-precision approximation

Attention can already be efficiently approximated to extremely high accuracy when the data dimension and norms are sufficiently small. Specifically, when either $d = \mathcal{O}(\log n)$ and $B = o(\sqrt{\log n})$, or $d = o(\log n)$ and $B = \mathcal{O}(\sqrt{\log n})$, there are algorithms with near-linear runtime solving $\mathsf{AbsAtt}(n, d, B, \eta_{\text{abs}})$ for any constant error guarantee $\eta_{\text{abs}} > 0$. Concretely, for any constant $c > 0$, Alman & Song (2023) achieve error vanishing as n^{-c} with the sequence length in $n^{1+o(1)}$ time, and Schröder & Mackey (2026) even achieve error vanishing as $n^{-\omega(1)}$ in $n^{1+o(1)}$ time. Although asymptotic, these guarantees amount effectively to exact attention computation at the sequence lengths for which approximate attention is practically relevant. Consequently, the mildest regime for which we do not already know how to solve Problem 1 efficiently is at the boundary where $d = \Theta(\log n)$ and $B = \Theta(\sqrt{\log n})$.

At this boundary, existing runtime lower bounds show that efficient algorithms can no longer guarantee extreme accuracy. Concretely, Alman & Song (2023) show that *some* polynomial error decay cannot be attained in truly subquadratic time unless SETH fails.

Theorem 3.4 (Alman & Song (2023, Thm. 4.6))**.** *Fix $q > 0$. Assuming SETH, there exist constants $D, B, C > 0$ such that $\mathsf{AbsAtt}(n, D\log n, B\sqrt{\log n}, n^{-C})$ cannot be solved in $\mathcal{O}(n^{2-q})$ time.*

While this result rules out the extreme accuracy possible for small d and B, it leaves open whether some of this accuracy can be exchanged for fast attention approximation in broader regimes. In particular, Theorem 3.4 does not preclude a linear-time algorithm for $\mathsf{AbsAtt}(n, \log n, \sqrt{\log n}, 0.49)$.

As summarized in Table 1, other lower bounds do not resolve this question either. Closing the gap would require ruling out coarse approximation when $d = \Theta(\log n)$ and $B = \Theta(\sqrt{\log n})$. Duman Keles et al. (2023) rule out nontrivial entrywise-relative approximation, but only for superpolynomially small additive error and under more restrictive assumptions on d and B. Likewise, Gupta et al. (2026, Thm. 5.7) applies to $d = \Theta(\log n)$ and $B = \Theta(\sqrt{\log n})$, but again only shows that some polynomial decay is impossible, while Gupta et al. (2026, Thm. 5.4) rules out constant-error approximation only in the practically unrealistic setting where query and key entries grow polynomially with n.

Consequently, existing lower bounds leave open whether truly subquadratic algorithms can provide uniform approximation guarantees beyond the parameter regime of known algorithms with near-linear runtime and near-exact approximation. Closing this gap requires ruling out the entire nontrivial range of additive and relative errors already at $d = \Theta(\log n)$ and $B = \Theta(\sqrt{\log n})$.

3.3 NONTRIVIAL GUARANTEES ARE IMPOSSIBLE UNDER THE MILDEST ASSUMPTIONS

We show that, once $d = \Theta(\log n)$ and $B = \Theta(\sqrt{\log n})$, no truly subquadratic algorithm can solve either $\mathsf{AbsAtt}(n, d, B, \eta_{\text{abs}})$ for any $\eta_{\text{abs}} < 1/2$ or $\mathsf{RelAtt}_p(n, d, B, \eta_{\text{rel}})$ for any $\eta_{\text{rel}} < 1$, unless SETH fails. To that end, we strengthen the proof of Alman & Song (2023), which draws on the connection between computing attention and finding close vectors.

Definition 3.5 (Closest pair problem)**.** *For $n, d \in \mathbb{N}$ and $\varepsilon > 0$, the problem $\mathsf{CP}(n, d, \varepsilon)$ is the following. Given sets $\{\boldsymbol{a}_1, \dots, \boldsymbol{a}_n\}, \{\boldsymbol{b}_1, \dots, \boldsymbol{b}_n\} \subseteq \{0,1\}^d$, find $i^\star, j^\star \in [n]$ such that*

$$\|\boldsymbol{a}_{i^\star} - \boldsymbol{b}_{j^\star}\|_0 \le (1+\varepsilon) \min_{i,j \in [n]} \|\boldsymbol{a}_i - \boldsymbol{b}_j\|_0.$$

The central idea underlying the proof of Alman & Song (2023) is that the attention matrix can be repurposed to encode pairwise distances between the two sets of vectors, while the subsequent value aggregation is used to recover a close pair. Hence, a fast algorithm for computing attention yields an essentially equally fast algorithm for finding an approximate closest pair. Finding such a pair, however among two sets of n binary vectors in dimension $d = \mathcal{O}(\log n)$ cannot be done much more efficiently than checking all n^2 possible pairs unless SETH fails, as Rubinstein (2018) proves.

Proposition 3.6 (Rubinstein (2018, Thm. 1.1))**.** *Fix $q > 0$. Assuming SETH, there exist $C > 0$ and $\varepsilon \in (0,1)$ such that $\mathsf{CP}(n, C \log n, \varepsilon)$ cannot be solved in $\mathcal{O}(n^{2-q})$ time.*

A truly subquadratic time algorithm for attention therefore cannot exist under SETH. When only a fast *approximate* algorithm for attention is available, however, the particular construction of Alman & Song (2023) recovers such a pair only when the approximation error is extremely small. We therefore strengthen their construction to find a close pair reliably even when only a very coarse approximation of attention is available. Put together, any truly subquadratic algorithm for attention with a nontrivial approximation guarantee, already for $d = \Theta(\log n)$ and $B = \Theta(\sqrt{\log n})$, then implies a truly subquadratic algorithm for the closest pair problem, which would contradict SETH via Proposition 3.6. The following theorem formalizes this result. We sketch the main idea here and give the full proof in Appendix A.

Theorem 3.7. *Fix $\eta_{\text{abs}} \in [0, \frac{1}{2})$, $\eta_{\text{rel}} \in [0, 1)$, and $q > 0$. Assuming SETH, there exist $C_d, C_B > 0$ such that neither $\mathsf{AbsAtt}(n, D, B, \eta_{\text{abs}})$ nor $\mathsf{RelAtt}_p(n, D, B, \eta_{\text{rel}})$ can be solved in $\mathcal{O}(n^{2-q})$ time, for $D \coloneqq C_d \log n$, $B \coloneqq C_B\sqrt{\log n}$, and any $p \in [1, \infty]$.*

Proof sketch. Given two sets of binary vectors as an input to the closest pair problem, we construct from them query and key matrices so that the full $n \times n$ attention matrix of all key–query comparisons already contains the information needed to find close pairs, with the largest attention scores corresponding to closest vectors. The difficulty is that an attention algorithm does not return these scores but only their normalized aggregation against the values, which can obscure the contribution of any individual close pair. Our main idea is to turn this normalization itself into a test whether a close pair with distance below a given threshold exists. To achieve this, we add dummy keys and augment the genuine queries and keys so that close pairs score above the dummies, while far pairs score below them. We then choose the values to measure the total attention paid to genuine keys. By scaling the logits appropriately, a single close genuine key can be made to outweigh all dummy keys whereas the combined contribution of all far genuine keys remains negligible, all while keeping the query and key entries bounded by $B = \mathcal{O}(\sqrt{\log n})$. The resulting attention output then becomes an almost-binary indicator of whether a pair below distance t exists, allowing even any fixed nontrivial additive or relative approximation to distinguish the two cases. Repeating this test over logarithmically many thresholds recovers an approximate closest pair. Thus, a truly subquadratic algorithm for approximating attention would yield a truly subquadratic algorithm for the closest pair problem, contradicting Proposition 3.6 unless SETH fails. □

Theorem 3.7 therefore settles the boundary of efficient attention approximation with uniform guarantees. At $d = \Theta(\log n)$ and $B = \Theta(\sqrt{\log n})$, even the weakest nontrivial guarantees require essentially quadratic work under standard complexity-theoretic assumptions. At the same time, relaxing either parameter to $d = o(\log n)$ or $B = o(\sqrt{\log n})$ admits algorithms with near-linear runtime and near-exact approximation (Alman & Song, 2023; Schröder & Mackey, 2026). Our results therefore identify a sharp transition of efficient attention approximation from possible at high accuracy to impossible at any nontrivial accuracy.

4 APPROXIMATION REMAINS HARD AFTER KV CACHE PREPROCESSING

Transformer inference in practice is typically optimized for *autoregressive generation*, where the model extends the original input sequence one token at a time. This differs from *prefill*, where the initial input is processed all at once. Because autoregressive generation is inherently sequential, it is much harder to parallelize than prefill, making attention during generation often much more costly than during prefill. This has motivated substantial work on accelerating attention specifically during generation through preprocessing of the keys and values produced during prefill, collectively known as the *KV cache*. In particular, such methods may spend additional computation during or after prefill to compress, index, or otherwise organize the KV cache to reduce the cost of subsequent attention queries (Li et al., 2024; Tang et al., 2024; Chen et al., 2025; Liu et al., 2025).

Allowing algorithms to preprocess the KV cache before approximating attention could, in principle, make uniform approximation substantially easier. Yet existing runtime lower bounds do not address this tradeoff between preprocessing and subsequent query time, despite its practical relevance. We therefore ask whether KV cache preprocessing enables fast attention approximation with nontrivial uniform guarantees that are impossible without it. To answer this question, Section 4.1 adapts the computational problem of approximate attention to autoregressive generation after KV cache preprocessing. Section 4.2 then shows that, even with arbitrary polynomial-time preprocessing and under the same mild assumptions as before, nontrivial uniform guarantees remain impossible in every regime not already covered by fast high-accuracy approximation algorithms.

4.1 PROBLEM FORMULATION

To model approximate attention during autoregressive generation, we separate the approximation problem into two stages. In the preprocessing stage, the algorithm receives the KV cache and constructs from it an arbitrary data structure. Only afterward is a single query revealed, and the algorithm must compute its attention output using the preprocessed cache.

Problem 2: Approximating attention with preprocessing

Fix parameters $n, d \in \mathbb{N}$ and $B \geq 0$. Given matrices $\boldsymbol{K} \in [-B, B]^{n \times d}$ and $\boldsymbol{V} \in [0, 1]^{n \times d}$, preprocess them. Then, given a query $\boldsymbol{Q} \in [-B, B]^{1 \times d}$, the task is to compute a matrix $\boldsymbol{T} \in \mathbb{R}^{1 \times d}$. For this output, the problem

$$\begin{aligned} &\mathsf{AbsPreAtt}(n, d, B, \eta_{\mathrm{abs}}) \quad \text{requires} \quad \|\boldsymbol{T} - \mathrm{Att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})\|_{\ell_\infty} \leq \eta_{\mathrm{abs}}, \quad \text{and} \\ &\mathsf{RelPreAtt}(n, d, B, \eta_{\mathrm{rel}}) \quad \text{requires} \quad \|\boldsymbol{T} - \mathrm{Att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})\|_{S_2} \leq \eta_{\mathrm{rel}} \|\mathrm{Att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})\|_{S_2}. \end{aligned}$$

The naive exact implementation for Problem 2 simply stores the KV cache and subsequently scans the entire cache for each query, costing $\mathcal{O}(nd)$ preprocessing time and $\mathcal{O}(nd)$ decoding time. Furthermore, as before, additive error $1/2$ and relative error 1 are trivial to guarantee without preprocessing. The relevant question is therefore whether nontrivial uniform approximation guarantees can be achieved in truly sublinear decoding time $\mathcal{O}(n^{1-q})$ after potentially substantial preprocessing.

4.2 NONTRIVIAL GUARANTEES ARE IMPOSSIBLE UNDER THE MILDEST ASSUMPTIONS

To rule out fast approximation even after substantial preprocessing, we again exploit the connection between attention and the closest pair problem from Definition 3.5. In fact, Rubinstein (2018, Cor. 1.3) already shows that finding an approximate closest pair still requires checking essentially all pairs even when one of the two vector sets is revealed in advance and may be preprocessed in polynomial time. Incidentally, the reduction underlying Theorem 3.7 already mirrors this structure: one vector set determines $\boldsymbol{K}$ and $\boldsymbol{V}$, while the other is used only to form $\boldsymbol{Q}$. The same reduction therefore naturally extends to Problem 2. We defer the proof details to Appendix A and state only the conclusion here. Under the same mild assumptions as before, arbitrary polynomial-time preprocessing does not enable efficient decoding with nontrivial uniform approximation guarantees.

Theorem 4.1. *Fix $\eta_{\mathrm{abs}} \in [0, \frac{1}{2})$, $\eta_{\mathrm{rel}} \in [0, 1)$, and $k, q > 0$. Assuming* SETH*, there exist $C_d, C_B > 0$ such that neither $\mathsf{AbsPreAtt}(n, D, B, \eta_{\mathrm{abs}})$ nor $\mathsf{RelPreAtt}(n, D, B, \eta_{\mathrm{rel}})$ can be solved with $\mathcal{O}(n^k)$ preprocessing time and $\mathcal{O}(n^{1-q})$ decoding time, for $D \coloneqq C_d \log n$ and $B \coloneqq C_B \sqrt{\log n}$.*

5 APPROXIMATION REMAINS HARD FOR SPARSE ATTENTION

Attention often exhibits sparse structure in practice, with a small number of tokens receiving most of the attention for any particular query. Many practical methods therefore exploit this structure to accelerate attention during generation by identifying the relevant tokens for each query and attending only over them (Li et al., 2024; Tang et al., 2024; Chen et al., 2025; Liu et al., 2025). Sparse attention methods therefore pursue an objective not captured by Problem 1 and 2: rather than approximating the full attention output, they aim to retain a small set of tokens receiving most of the attention. This can preserve accurate performance at substantially reduced cost if those tokens can be identified efficiently.

Identifying the important tokens could, in principle, be easier than approximating the full attention output, and neither prior work nor our preceding lower bounds rule out this possibility. This section therefore asks whether such retrieval is possible with uniform guarantees over all inputs. To that end, Section 5.1 formalizes retrieval under sparsity as a computational problem. Section 5.2 then shows that, even after arbitrary polynomial-time preprocessing of the KV cache and under the same mild assumptions as before, no fast algorithm can identify a small set of tokens receiving substantial attention for a given query, even when its attention distribution is highly concentrated.

5.1 PROBLEM FORMULATION

A sparse attention method maps a query to a small subset of relevant keys and subsequently evaluates attention only over this subset. To preserve the relevant information from the full computation, the selected keys should collectively receive a substantial fraction of the total attention. We therefore allow an algorithm to preprocess the keys into an arbitrary data structure and, once a query is revealed, ask it to return a small subset capturing a constant fraction of its attention. This objective is meaningful only when the attention distribution itself exhibits sparse structure. If all keys are identical, for example, attention is uniform and no small subset can receive much of the attention. We therefore restrict the problem to queries for which the attention distribution is actually sparse by promising that some key already receives a constant fraction of the attention. Returning any small set containing such a key then suffices.

Problem 3: Sparse attention with preprocessing

Fix parameters $n, d \in \mathbb{N}$, $B \geq 0$, and $\alpha \in (0, 1)$. Given a matrix $\boldsymbol{K} \in [-B, B]^{n \times d}$, preprocess it. Then, given a query $\boldsymbol{q} \in [-B, B]^d$, the problem $\mathsf{SparseAtt}(n, d, B, \alpha)$ asks for a subset $\mathbb{I} \subseteq [n]$ of the keys receiving attention

$$\textstyle\sum_{i \in \mathbb{I}} \mathrm{softmax}_i(\boldsymbol{K}\boldsymbol{q}/d) \geq \alpha \quad \text{promised that} \quad \mathrm{softmax}_i(\boldsymbol{K}\boldsymbol{q}/d) \geq \alpha \quad \text{for some } i \in [n].$$

To solve this problem, a trivial algorithm could compute all attention scores and return a singleton containing the promised key, or it could return all keys, which always captures the full attention. Both approaches require $\mathcal{O}(n)$ time, however, and therefore provide no speedup over simply computing attention over all keys. The relevant question is therefore whether Problem 3 can be solved in truly sublinear time $\mathcal{O}(n^{1-q})$ after preprocessing. In particular, we count the size of the set toward the runtime, requiring it to carry truly fewer than $\mathcal{O}(n)$ keys.

5.2 EFFICIENTLY RETAINING TOKENS RECEIVING SUBSTANTIAL ATTENTION IS IMPOSSIBLE

We show that, under the same mild assumptions as before, no truly sublinear algorithm can retain any constant fraction of the attention after arbitrary polynomial-time preprocessing. While we still exploit the connection between attention and the closest pair problem from Definition 3.5, the reduction of Alman & Song (2023) underlying Theorems 3.7 and 4.1 does not extend directly to sparse attention. Its construction encodes close pairs through attention scores that are large relative to those of far pairs. Yet, a closest pair instance may contain many approximately closest pairs at similar distances. On such inputs, approximating the full attention output can reveal *some* such pair even when the attention distribution is diffuse, violating the promise of Problem 3. To obtain hardness in the presence of sparsity, we therefore modify the reduction so that one approximately closest candidate dominates all others. We achieve this by randomly perturbing the encoded keys in a way that isolates

one of the approximately closest keys to receive almost all of the attention. Any subset capturing a constant fraction of the attention must then contain this key and thereby reveal an approximate closest pair. We state the resulting lower bound below and sketch the reduction, deferring the full proof to Appendix A.

Theorem 5.1. *Fix any $\alpha \in (0,1)$ and $k, q > 0$. Assuming* SETH*, there exist $C_d, C_B > 0$ such that* $\mathsf{SparseAtt}(n, D, B, \alpha)$ *cannot be solved with $\mathcal{O}(n^k)$ preprocessing time and $\mathcal{O}(n^{1-q})$ decoding time, for $D \coloneqq C_d \log n$ and $B \coloneqq C_B\sqrt{\log n}$.*

Proof sketch. Suppose there were an algorithm for Problem 3 with polynomial preprocessing time and truly sublinear decoding time. We use it to construct a bounded-error randomized algorithm for approximate nearest-neighbor search after polynomial preprocessing, contradicting SETH under the lower bound of Rubinstein (2018). As in Theorem 3.7, we encode the preprocessed vectors as keys and the later query vector as an attention query, so that nearby vectors receive much larger scores than distant ones. The difficulty is that there may be many close keys, resulting in a diffuse attention distribution, whereas a sparse attention algorithm requires the attention distribution to be concentrated. Our main idea is to randomly perturb the key matrix so that one close key dominates all others with high probability. Concretely, we create a small number of copies of the key matrix and randomly shift each key by an exponentially distributed amount in a fixed direction. We truncate these shifts so that all entries remain bounded by $B = \mathcal{O}(\sqrt{\log n})$ and far vectors continue to receive negligible attention. We show that, with high probability, at least one perturbed copy contains a key corresponding to an approximate closest neighbor of the query while receiving at least $1 - \frac{1}{n} > 1 - \alpha$ attention. For that copy, any subset capturing an α-fraction of the attention must contain this key, thereby revealing an approximate closest neighbor. Running the hypothetical sparse attention algorithm on all copies still yields a truly sublinear bounded-error randomized algorithm for the closest-pair search problem and thus contradicting SETH. □

This demonstrates that retrieval of relevant tokens under sparsity is fundamentally hard to guarantee uniformly over all inputs: in the mildest regime beyond the reach of highly accurate attention approximation with near-linear runtime, no algorithm can uniformly and efficiently identify a small set of tokens receiving substantial attention under future queries, even when attention is highly concentrated and even when allowing arbitrary polynomial-time preprocessing of the KV cache. Consequently, efficient sparse attention must exploit additional structure of the inputs beyond sparsity itself.

6 Conclusion

Motivated by the computational cost of Transformers, we study whether attention can be approximated efficiently with uniform guarantees over all inputs. Under standard complexity-theoretic assumptions, we identify that the known tractable regimes mark the limit of what is possible: outside the regimes of existing algorithms with near-linear runtime and vanishingly small error, approximating attention with a nontrivial additive or relative guarantee uniformly over all inputs requires essentially quadratic time. Interestingly, this barrier persists during autoregressive generation after arbitrary polynomial-time preprocessing of the KV cache, and then even for retrieval under sparsity. While our results make substantial improvements in algorithms with uniform worst-case guarantees unlikely, they do not rule out algorithms that perform well empirically or obtain strong guarantees under structural assumptions on the inputs. In fact, our lower bounds substantiate the need for such assumptions: beyond the already tractable regimes, exploiting input structure is genuinely necessary to obtain meaningful guarantees for fast approximate attention.

AI USE STATEMENT

In the preparation of this paper, large language models were used to polish the authors' original writing. All scientific content was written and reviewed by the authors.

REFERENCES

Maryam Aliakbarpour, Vladimir Braverman, Junze Yin, and Haochen Zhang. Support basis: Fast attention beyond bounded entries. In *International Conference on Artificial Intelligence and Statistics*, volume 300, pp. 325–333, 2026. URL https://proceedings.mlr.press/v300/aliakbarpour26a.html.

Josh Alman and Zhao Song. Fast attention requires bounded entries. In *Advances in Neural Information Processing Systems*, volume 36, pp. 63117–63135, 2023. doi: 10.52202/075280-2755.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. URL https://arxiv.org/abs/2004.05150.

Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, and Christopher Ré. Scatterbrain: Unifying sparse and low-rank attention. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17413–17426, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/9185f3ec501c674c7c788464a36e7fb3-Abstract.html.

Zhuoming Chen, Ranajoy Sadhukhan, Zihao Ye, Yang Zhou, Jianyu Zhang, Niklas Nolte, Yuandong Tian, Matthijs Douze, Leon Bottou, Zhihao Jia, and Beidi Chen. MagicPIG: LSH sampling for efficient LLM generation. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/6d50d824ae819d5a961c1d8edc15e833-Paper-Conference.pdf.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. URL https://arxiv.org/abs/1904.10509.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Łukasz Kaiser, David Benjamin Belanger, Lucy J. Colwell, and Adrian Weller. Rethinking attention with Performers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/98ed250b203d1ac6b24bbcf263e3d4a7-Paper-Conference.pdf.

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)*, pp. 4171–4186, 2019. doi: 10.18653/v1/N19-1423.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. URL https://arxiv.org/abs/2010.11929.

Feyza Duman Keles, Pruthuvi Mahesakya Wijewardena, and Chinmay Hegde. On the computational complexity of self-attention. In *Proceedings of The 34th International Conference on Algorithmic Learning Theory*, volume 201 of *Proceedings of Machine Learning Research*, pp. 597–619. PMLR, 2023. URL https://proceedings.mlr.press/v201/duman-keles23a.html.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech 2020*, pp. 5036–5040, 2020. doi: 10.21437/Interspeech.2020-3015.

Shreya Gupta, Boyang Huang, Barna Saha, Yinzhan Xu, and Christopher Ye. Subquadratic algorithms and hardness for attention with any temperature. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/8a01099096c85890b1d1aff3c6b4ea56-Paper-Conference.pdf.

Insu Han, Rajesh Jayaram, Amin Karbasi, Vahab Mirrokni, David Woodruff, and Amir Zandieh. HyperAttention: Long-context attention in near-linear time. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/ab5aa940590399350401c57cdf52ce78-Paper-Conference.pdf.

Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi: 10.1006/jcss.2000.1727.

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

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 22947–22970, 2024. doi: 10.52202/079017-0722.

Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. FQ-ViT: Post-training quantization for fully quantized vision transformer. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 1173–1179, 2022. doi: 10.24963/ijcai.2022/164.

Di Liu, Meng Chen, Baotong Lu, Huiqiang Jiang, Zhenhua Han, Qianxi Zhang, Qi Chen, Chengruidong Zhang, Bailu Ding, Kai Zhang, Chen Chen, Fan Yang, Yuqing Yang, and Lili Qiu. Retrievalattention: Accelerating long-context LLM inference via vector retrieval. In *Advances in Neural Information Processing Systems*, volume 38, pp. 54358–54385, 2025. doi: 10.52202/085713-1816.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32332–32344. PMLR, 2024. URL https://proceedings.mlr.press/v235/liu24bz.html.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32, pp. 14014–14024, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/2c601ad9d2ff9bc8b282670cdd54f69f-Abstract.html.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021. doi: 10.1073/pnas.2016239118.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021. doi: 10.1162/tacl_a_00353.

Aviad Rubinstein. Hardness of approximate nearest neighbor search. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1260–1268, 2018. doi: 10.1145/3188745.3188916.

Tobias Schröder and Lester Mackey. Wildcat: Near-linear attention in theory and practice. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=lfqyLp4hZm.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. QUEST: Query-aware sparsity for efficient long-context LLM inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 47901–47911. PMLR, 2024. URL https://proceedings.mlr.press/v235/tang24l.html.

Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse Sinkhorn attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9438–9447. PMLR, 2020. URL https://proceedings.mlr.press/v119/tay20a.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. Fast transformers with clustered attention. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21665–21674, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f6a8dd1c954c8506aadc764cc32b895e-Abstract.html.

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

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A Nyström-based algorithm for approximating self-attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14138–14148, 2021. doi: 10.1609/aaai.v35i16.17664.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big Bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17283–17297, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c8512d142a2d849725f31a9a7a361ab9-Abstract.html.

Amir Zandieh, Insu Han, Majid Daliri, and Amin Karbasi. KDEformer: Accelerating transformers via kernel density estimation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40605–40623. PMLR, 2023. URL https://proceedings.mlr.press/v202/zandieh23a.html.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325.

A PROOFS

This appendix contains the formal proofs of Theorems 3.7, 4.1 and 5.1. Section A.1 presents a variant of the approximate closest pair problem, to which all our results reduce, and connects it to the lower bounds of Rubinstein (2018). Section A.2 shows how even a coarse approximation of attention can be used to identify approximate closest pairs, which Section A.3 uses to prove Theorems 3.7 and 4.1. Similarly, Section A.4 shows how sparse attention retrieval can be used to identify approximate closest pairs, which Section A.5 uses to prove Theorem 5.1.

A.1 SOURCE PROBLEM

We prove all our results by showing that a fast algorithm for attention with a given approximation guarantee would ultimately give a fast algorithm for the approximate closest pairs problem below, which Rubinstein (2018) proves is impossible unless SETH fails.

Definition 3.5 (Closest pair problem)**.** *For $n, d \in \mathbb{N}$ and $\varepsilon > 0$, the problem $\mathsf{CP}(n, d, \varepsilon)$ is the following. Given sets $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}, \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\} \subseteq \{0,1\}^d$, find $i^\star, j^\star \in [n]$ such that*

$$\|\boldsymbol{a}_{i^\star} - \boldsymbol{b}_{j^\star}\|_0 \leq (1+\varepsilon) \min_{i,j \in [n]} \|\boldsymbol{a}_i - \boldsymbol{b}_j\|_0.$$

Proposition 3.6 (Rubinstein (2018, Thm. 1.1))**.** *Fix $q > 0$. Assuming SETH, there exist $C > 0$ and $\varepsilon \in (0,1)$ such that $\mathsf{CP}(n, C \log n, \varepsilon)$ cannot be solved in $\mathcal{O}(n^{2-q})$ time.*

Remark A.1. *Although Proposition 3.6, as stated, only concerns deterministic algorithms it also holds against bounded-error randomized algorithms under randomized SETH. Indeed, both the standard reduction from k-SAT to Orthogonal Vectors and the subsequent reduction of Rubinstein (2018) to CP are deterministic. Consequently, a bounded-error randomized algorithm for CP with truly subquadratic runtime would contradict SETH as stated in Hypothesis 3.3.*

For convenience, we define the decision variant of this problem with preprocessing. We also follow Alman & Song (2023) in working with inputs of balanced Hamming weights, that is, vectors in dimension $2d$ that have Hamming weight d. To that end, define, for even $d \in \mathbb{N}$, the slice $\mathbb{M}_d \coloneqq \{\boldsymbol{v} \in \{0,1\}^d \mid \|\boldsymbol{v}\|_0 = d/2\}$ of the Boolean cube with balanced Hamming weight.

Definition A.2 (Preprocessed Gap Closest Pair)**.** *For $n, m, d \in \mathbb{N}$ and $\varepsilon > 0$, the problem $\mathsf{PreCP}(n, m, d, \varepsilon)$ is the following. Given a threshold $t \in [0, d]$ and set $\mathbb{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\} \subseteq \mathbb{M}_d$, preprocess them. Then, given vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m \in \mathbb{M}_d$, decide if*

$$\min_{i \in [m],\ j \in [n]} \|\boldsymbol{a}_i - \boldsymbol{b}_j\|_0 \leq t \qquad \textit{or} \qquad \min_{i \in [m],\ j \in [n]} \|\boldsymbol{a}_i - \boldsymbol{b}_j\|_0 > (1+\varepsilon)t,$$

promised that one of the two holds.

Solving this problem much more efficiently than checking all mn possible pairs remains hard even for bounded-error randomized algorithm with polynomial preprocessing and under the restriction to balanced inputs, as the following proposition shows. The proof effectively combines the arguments behind Rubinstein (2018, Cor. 1.3) and Alman & Song (2023, Rem. 4.4) together with a standard boosting trick for randomization.

Proposition A.3 (Balanced gap version of Rubinstein (2018, Cor. 1.3))**.** *Fix $k, q > 0$. Suppose that SETH holds. Then, there exist $C, \varepsilon > 0$ such that $\mathsf{PreCP}(n, m(n), d, \varepsilon)$ cannot be solved, not even by a bounded-error randomized algorithm, with $\mathcal{O}(n^k)$ preprocessing time and $\mathcal{O}(m(n)n^{1-q})$ query time, for $d \coloneqq C \log n$ and any function $m : \mathbb{N} \to \mathbb{N}$ with $m(n) \in [2n]$.*

Proof. Assume w.l.o.g. that $k > 1$ and $q \leq 1/4$. Let $\gamma \coloneqq 1/(2k)$ and $\bar{q} \coloneqq q\gamma/2$. By Proposition 3.6, there exist $C > 0$ and $\varepsilon \in (0,1)$ such that $\mathsf{CP}(n, \frac{\gamma C}{2} \log n, \varepsilon)$ cannot be solved in $\mathcal{O}(n^{2-\bar{q}})$ time. Suppose, for contradiction, that $\mathsf{PreCP}(n, m(n), d, \varepsilon)$ can be solved by a bounded-error randomized algorithm, for some function $m : \mathbb{N} \to \mathbb{N}$ with $m(n) \in [2n]$, with $\mathcal{O}(n^k)$ preprocessing time and $\mathcal{O}(m(n)n^{1-q})$ query time. We construct an algorithm for $\mathsf{CP}(n, \gamma d/2, \varepsilon)$ running in $\mathcal{O}(n^{2-\bar{q}})$ time, contradicting Proposition 3.6. To that end, let sets $\mathbb{A} = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\} \subseteq \{0,1\}^{\gamma d/2}$ and $\mathbb{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\} \subseteq \{0,1\}^{\gamma d/2}$ be inputs to $\mathsf{CP}(n, \frac{\gamma C}{2} \log n, \varepsilon)$. We replace them with the balanced sets $\widehat{\mathbb{A}} \coloneqq \{\widehat{\boldsymbol{a}}_1, \ldots, \widehat{\boldsymbol{a}}_n\} \subseteq \mathbb{M}_{\gamma d}$ and $\widehat{\mathbb{B}} \coloneqq \{\widehat{\boldsymbol{b}}_1, \ldots, \widehat{\boldsymbol{b}}_n\} \subseteq \mathbb{M}_{\gamma d}$, where $\widehat{\boldsymbol{x}} \coloneqq (\boldsymbol{x}, \mathbf{1} - \boldsymbol{x}) \in \mathbb{M}_{\gamma d}$, so that $\|\boldsymbol{a} - \boldsymbol{b}\|_0 = \frac{1}{2}\|\widehat{\boldsymbol{a}} - \widehat{\boldsymbol{b}}\|_0$.

We now devise the algorithm. Let $r \coloneqq n^\gamma$ and $s \coloneqq m(r)$ so that $\gamma d = C \log r$. Partition $\widehat{\mathbb{B}}$ into sets $\widehat{\mathbb{B}}_1, \dots, \widehat{\mathbb{B}}_{\lceil n/r \rceil}$, each of size at most r. Then, for each threshold $t \in \{0, \dots, \gamma d\}$ in increasing order, do the following. Apply the assumed preprocessing algorithm for PreCP independently $L \coloneqq C_L \log n$ times to every set $\widehat{\mathbb{B}}_j$ separately, padding it to r separate vectors by duplication when necessary. Additionally, partition $\widehat{\mathbb{A}}$ into sets $\widehat{\mathbb{A}}_1, \dots, \widehat{\mathbb{A}}_{\lceil n/s \rceil}$, each of size at most s. On each preprocessed subset $\widehat{\mathbb{B}}_j$, independently run the corresponding L decoding algorithms on each query bulk $\widehat{\mathbb{A}}_i$, again padding them to $s = m(r)$ vectors when necessary, and use their majority answer. That is, accept if at least half of the independent invocations of the decoding algorithm accept.

Choosing C_L as a sufficiently large constant guarantees that, on every promised input, each majority answer is incorrect with probability at most n^{-4}. Overall, we compute at most $\mathcal{O}((\gamma d + 1)(n/r)(n/s)) = \mathcal{O}(n^2 \log n)$ such answers. Hence, by a union bound, with probability at least

$$1 - \mathcal{O}(n^2 \log n) n^{-4} \geq \frac{2}{3},$$

all majority answers on promised inputs are correct simultaneously. Condition on that event and consider the smallest t for which the majority answer accepts on some $\widehat{\mathbb{B}}_j$ and $\widehat{\mathbb{A}}_i$. Then,

$$\min_{\widehat{\boldsymbol{a}} \in \widehat{\mathbb{A}}_i,\, \widehat{\boldsymbol{b}} \in \widehat{\mathbb{B}}_j} \|\widehat{\boldsymbol{a}} - \widehat{\boldsymbol{b}}\|_0 \leq (1+\varepsilon) t \leq (1+\varepsilon) \min_{\widehat{\boldsymbol{a}} \in \widehat{\mathbb{A}}, \widehat{\boldsymbol{b}} \in \widehat{\mathbb{B}}} \|\widehat{\boldsymbol{a}} - \widehat{\boldsymbol{b}}\|_0,$$

where the first inequality uses that $\min_{\widehat{\boldsymbol{a}} \in \widehat{\mathbb{A}}_i, \widehat{\boldsymbol{b}} \in \widehat{\mathbb{B}}_j} \|\widehat{\boldsymbol{a}} - \widehat{\boldsymbol{b}}\|_0 > (1+\varepsilon)t$ would force the decoding algorithm to reject by the problem definition, and the second inequality uses that the algorithm must accept once $t = \min_{\widehat{\boldsymbol{a}} \in \widehat{\mathbb{A}}, \widehat{\boldsymbol{b}} \in \widehat{\mathbb{B}}} \|\widehat{\boldsymbol{a}} - \widehat{\boldsymbol{b}}\|_0$. Consequently, computing $\arg\min_{\widehat{\boldsymbol{a}} \in \widehat{\mathbb{A}}_i, \widehat{\boldsymbol{b}} \in \widehat{\mathbb{B}}_j}$ gives a pair $(\boldsymbol{a}, \boldsymbol{b})$ for the original input with

$$\|\boldsymbol{a} - \boldsymbol{b}\|_0 = \frac{1}{2}\|\widehat{\boldsymbol{a}} - \widehat{\boldsymbol{b}}\|_0 \leq \frac{1}{2}(1+\varepsilon) \min_{\widehat{\boldsymbol{a}}' \in \widehat{\mathbb{A}}, \widehat{\boldsymbol{b}}' \in \widehat{\mathbb{B}}} \|\widehat{\boldsymbol{a}}' - \widehat{\boldsymbol{b}}'\|_0 = (1+\varepsilon) \min_{\boldsymbol{a}' \in \mathbb{A}, \boldsymbol{b}' \in \mathbb{B}} \|\boldsymbol{a}' - \boldsymbol{b}'\|_0$$

satisfying the output condition of CP. Since we only need to search the small subsets $\widehat{\mathbb{A}}_i$ and $\widehat{\mathbb{B}}_j$, such a pair can be computed by brute force in $\mathcal{O}(rsd) \leq \mathcal{O}(r^2 d) \leq \mathcal{O}(n \log n)$ time.

Consequently, we obtain an algorithm with bounded error for $\mathsf{CP}(n, \gamma d/2, \varepsilon)$. For each threshold, the preprocessing takes $\mathcal{O}(L \frac{r^k n}{r}) \leq \mathcal{O}(n^{3/2} \log n)$ time and the decoding takes $\mathcal{O}(L \frac{n}{r} \frac{n}{s} s r^{1-q}) \leq \mathcal{O}(n^{2-q\gamma} \log n)$ time. Repeating for all $\gamma d + 1$ thresholds adds a factor $\mathcal{O}(d) = \mathcal{O}(\log n)$. Thus, the algorithm runs in $\mathcal{O}(n^{2-q\gamma/2})$ time, contradicting Proposition 3.6 together with Remark A.1. □

We also use the fact that the problem is easy to solve when the decision threshold t is small. To prove this, we apply the argument of Alman & Song (2023)[Lem. B.1, Case 1] to our particular problem formulation to give a brute-force algorithm searching the immediate vicinity of every preprocessed vector, resulting in $\mathcal{O}(mn^{1-q})$ runtime when the search space is sufficiently small.

Lemma A.4. *Fix $C_d > 0$ and $q \in (0, 1)$. There exists a constant $c \in (0, C_d/2)$ such that $\mathsf{PreCP}(n, m, C_d \log n, \varepsilon)$ can be solved with $\mathcal{O}(nd)$ preprocessing time and $\mathcal{O}(mn^{1-q})$ decoding time whenever the input threshold t satisfies $t \leq c \log n$.*

Proof. Let $d \coloneqq C_d \log n$. Choose any $c \in (0, C_d/2)$ such that $c \log\left(\frac{eC_d}{c}\right) < 1 - q$. Such a choice is always possible since $\lim_{c \searrow 0} c \log\left(\frac{eC_d}{c}\right) = 0$. Now, let $\mathbb{B} \coloneqq \{\boldsymbol{b}_1, \dots, \boldsymbol{b}_n\} \subseteq \mathbb{M}_d$ be the input set to $\mathsf{PreCP}(n, m, d, \varepsilon)$ and $\boldsymbol{a}_1, \dots, \boldsymbol{a}_m \in \mathbb{M}_d$ the query revealed after preprocessing. We give a brute-force algorithm for deciding this input, following the Hamming-ball enumeration of Alman & Song (2023, Lem. B.1, Case 1). Preprocess $\mathbb{B}$ by storing it in a trie. This takes $\mathcal{O}(nd)$ time. Then, enumerate, for all $i \in [m]$ separately, all vectors $\boldsymbol{b} \in \{0,1\}^d$ satisfying $\|\boldsymbol{a}_i - \boldsymbol{b}\|_0 \leq t$, and accept if any such $\boldsymbol{b}$ belongs to $\mathbb{B}$. This correctly decides the instance: if some $\boldsymbol{a}_i$ has a close partner in $\mathbb{B}$, exhaustive enumeration finds it, and if no $\boldsymbol{a}_i$ has such a partner, no partner is found. Since $t \leq c \log n < d/2$, the decoding time is

$$\mathcal{O}\left(md \sum_{k=0}^{\lfloor t \rfloor} \binom{d}{k}\right) \leq \mathcal{O}\left(md \left(\frac{ed}{t}\right)^t\right) \leq \mathcal{O}\left(m \log n \left(\frac{eC_d \log n}{c \log n}\right)^{c \log n}\right)$$
$$\leq \mathcal{O}\left(mn^{c \log(eC_d/c) + o(1)}\right) \leq \mathcal{O}\left(mn^{1-q}\right). \qquad \square$$

A.2 Finding close pairs from a coarse attention approximation

Following the general strategy of Alman & Song (2023), we encode a closest pair instance as an attention instance so that approximating the attention output reveals an approximate closest pair. Deviating from their construction, however, we ensure that recovering such a pair does not require a high-precision attention approximation. To achieve this, we shift the attention scores relative to a collection of dummy keys so that the contribution of a close pair dominates the output, while the combined contribution of all far pairs remains negligible. Importantly, the keys and values depend only on one of the two closest pair sets, whereas the queries depend only on the other. The construction is therefore compatible with preprocessing the keys and values before the query vectors are revealed. The following lemma formalizes these properties.

Lemma A.5. *Fix $\varepsilon, c, C_d > 0$. There exists $C_B > 0$ such that the following holds for all sufficiently large $n \in \mathbb{N}$ and $N \coloneqq 2n$, $d \coloneqq 2C_d \log n$, $D \coloneqq 4C_d \log N$, and $B \coloneqq C_B\sqrt{\log N}$.*

Choose $t \geq c \log n$ with $d \geq (1+\varepsilon)t$, a set of vectors $\mathbb{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\} \subseteq \mathbb{M}_d$, and a sequence $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m \in \mathbb{M}_d$. One can construct matrices $\boldsymbol{K} \in [-B, B]^{N \times D}$ and $\boldsymbol{V} \in [0,1]^{N \times D}$ independent of $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m$ and in $\mathcal{O}(nD)$ time, and another matrix $\boldsymbol{Q} \in [-B, B]^{m \times D}$ in $\mathcal{O}(mD)$ time such that $\mathrm{Att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{y}\boldsymbol{e}_1^\top$ for some $\boldsymbol{y} \in \mathbb{R}^m$ satisfying, for every $i \in [m]$,

$$y_i \geq \frac{n}{n+1} \qquad \text{if} \quad \min_{j \in [n]} \|\boldsymbol{a}_i - \boldsymbol{b}_j\|_0 \leq t, \quad \text{and}$$
$$y_i \leq \frac{1}{n+1} \qquad \text{if} \quad \min_{j \in [n]} \|\boldsymbol{a}_i - \boldsymbol{b}_j\|_0 \geq (1+\varepsilon)t.$$

Proof. Define

$$\rho \coloneqq \sqrt{D/(2d)}, \qquad \beta \coloneqq \frac{12d \log n}{\varepsilon t}, \qquad \text{and} \qquad \mu \coloneqq \frac{1}{2} - \left(\frac{1}{2} + \frac{\varepsilon}{3}\right)\frac{t}{d}$$

and note that

$$0 \leq \frac{\varepsilon}{6(1+\varepsilon)} \leq \mu \leq \frac{1}{2}$$

since $t/d \leq 1/(1+\varepsilon)$ by assumption.

We construct the key and value matrices as

$$\boldsymbol{K} \coloneqq \rho\sqrt{\beta} \begin{bmatrix} \boldsymbol{b}_1^\top & -\mu\boldsymbol{1}_d^\top & \boldsymbol{0}_{D-2d}^\top \\ \vdots & \vdots & \vdots \\ \boldsymbol{b}_n^\top & -\mu\boldsymbol{1}_d^\top & \boldsymbol{0}_{D-2d}^\top \\ \boldsymbol{0}_d^\top & \boldsymbol{0}_d^\top & \boldsymbol{0}_{D-2d}^\top \\ \vdots & \vdots & \vdots \\ \boldsymbol{0}_d^\top & \boldsymbol{0}_d^\top & \boldsymbol{0}_{D-2d}^\top \end{bmatrix} \in \mathbb{R}^{N \times D} \qquad \text{and} \qquad \boldsymbol{V} \coloneqq \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{N \times D}$$

and the query matrix as

$$\boldsymbol{Q} \coloneqq \rho\sqrt{\beta} \begin{bmatrix} \boldsymbol{a}_1^\top & \boldsymbol{1}_d^\top & \boldsymbol{0}_{D-2d}^\top \\ \vdots & \vdots & \vdots \\ \boldsymbol{a}_m^\top & \boldsymbol{1}_d^\top & \boldsymbol{0}_{D-2d}^\top \end{bmatrix} \in \mathbb{R}^{m \times D}.$$

Next, we choose C_B so that the constructed matrices satisfy the required entry bounds. Since $t \geq c \log n$, $d = 2C_d \log n$, and $\rho^2 = \log(2n)/\log n \leq 2$,

$$\rho^2\beta = \frac{\log N}{\log n}\frac{12d \log n}{\varepsilon t} \leq \frac{24C_d \log N}{\varepsilon c}$$

Hence, there exists a constant $C_B > 0$, depending only on C_d, ε, and c, such that $\rho\sqrt{\beta} \leq C_B\sqrt{\log N}$. Since $0 \leq \mu \leq 1/2$, it follows that $\|\boldsymbol{Q}\|_{\ell_\infty} \leq B$ and $\|\boldsymbol{K}\|_{\ell_\infty} \leq B$.

We now show that the attention output separates close from far pairs as claimed. To that end, let $\boldsymbol{Y} \coloneqq \mathrm{Att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})$. For $i \in [m]$ and $j \in [n]$, the logit between the i-th query and the j-th non-dummy key is

$$\begin{aligned}\frac{\langle \boldsymbol{Q}_i, \boldsymbol{K}_j \rangle}{D} &= \frac{\rho^2 \beta}{D} \left(\langle a_i, b_j \rangle - \mu d \right) = \frac{\beta}{2d} \left(\langle a_i, b_j \rangle - \mu d \right) \\ &= \frac{\beta}{2d} \left(\frac{\|a_i\|_2^2 + \|b_j\|_2^2 - \|a_i - b_j\|_2^2}{2} - \mu d \right) \\ &= \frac{\beta}{2d} \left(\frac{d - \|a_i - b_j\|_0}{2} - \mu d \right) \\ &= \frac{\beta}{4} \left(1 - \frac{\|a_i - b_j\|_0}{d} \right) - \frac{\beta \mu}{2}, \end{aligned} \tag{1}$$

where the third equality uses that a_i and b_j both have Hamming weight $d/2$. For $i \in [m]$, let

$$s_i \coloneqq \sum_{j=1}^{n} \exp\left(\frac{\langle \boldsymbol{Q}_i, \boldsymbol{K}_j \rangle}{D} \right) \tag{2}$$

be the total score assigned to non-dummy keys. By the definition of $\boldsymbol{V}$, the i-th attention output is

$$\boldsymbol{Y}_i = \frac{s_i}{s_i + n} \boldsymbol{e}_1^\top, \tag{3}$$

since the n dummy keys are 0 so that each of them has logit 0 and contributes score 1 and only the first column of $\boldsymbol{V}$ is nonzero. Now, fix $i \in [m]$ and distinguish the following two cases from the claim. Suppose first that $\min_{j \in [n]} \|a_i - b_j\|_0 \le t$. Then, Equation (1) and the definition of μ imply

$$\frac{\langle \boldsymbol{Q}_i, \boldsymbol{K}_j \rangle}{D} \ge \frac{\beta}{4} \left(1 - \frac{t}{d} \right) - \frac{\beta \mu}{2} = \frac{\beta \varepsilon t}{6d} = 2 \log(n)$$

for some $j \in [n]$. Consequently, $s_i \ge n^2$ and Equation (3) gives $Y_{i,1} \ge n/(n+1)$. Now, suppose instead that $\min_{j \in [n]} \|a_i - b_j\|_0 \ge (1+\varepsilon)t$. Then, Equation (1) and the definition of μ imply

$$\frac{\langle \boldsymbol{Q}_i, \boldsymbol{K}_j \rangle}{D} \le \frac{\beta}{4} \left(1 - \frac{(1+\varepsilon)t}{d} \right) - \frac{\beta \mu}{2} = -\frac{\beta \varepsilon t}{12d} = -\log n$$

for all $j \in [n]$. Consequently, $s_i \le 1$, and Equation (3) gives $Y_{i,1} \le 1/(n+1)$ for all $i \in [m]$.

Finally, $\boldsymbol{K}$ and $\boldsymbol{V}$ can be constructed in $\mathcal{O}(ND)$ time and $\boldsymbol{Q}$ in $\mathcal{O}(mD)$ time by copying and padding the respective inputs. That proves the claim. □

A.3 Proofs of Theorems 3.7 and 4.1

Theorems 3.7 and 4.1 can now be proved by using Lemma A.5 to encode an input to the closest pair problem from Definition A.2 as an attention input whose output gives an almost-binary indicator of whether an approximate close pair exists. Consequently, a fast algorithm with even a coarse approximation would give a fast algorithm for deciding the closest pair problem, which would contradict Proposition A.3. While the proof is essentially the same for both theorems, the underlying problem formulations differ slightly. For compactness, we therefore formulate below a generalized problem formulation including both Problem 1 and 2 as special cases. We then prove that this generalized problem cannot be solved much more efficiently than by the trivial algorithm for attention, from which Theorems 3.7 and 4.1 follow immediately.

Problem 4: Approximating attention with preprocessing (batch variant)
Fix parameters $n, m, d \in \mathbb{N}$ and $B \geq 0$. Given matrices $\boldsymbol{K} \in [-B, B]^{n \times d}$ and $\boldsymbol{V} \in [0,1]^{n \times d}$, preprocess them. Then, given a query $\boldsymbol{Q} \in [-B, B]^{m \times d}$, the task is to compute a matrix $\boldsymbol{T} \in \mathbb{R}^{m \times d}$. For this output and the exact target $\boldsymbol{Y} \coloneqq \mathrm{Att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})$, the problem

$$\begin{aligned} &\mathsf{AbsPreBatchAtt}(n, m, d, B, \eta_{\mathrm{abs}}) && \text{requires} && \|\boldsymbol{T} - \boldsymbol{Y}\|_{\ell_\infty} \leq \eta_{\mathrm{abs}}, \quad \text{and} \\ &\mathsf{RelPreBatchAtt}_p(n, m, d, B, \eta_{\mathrm{rel}}) && \text{requires} && \|\boldsymbol{T} - \boldsymbol{Y}\|_{\mathcal{S}_p} \leq \eta_{\mathrm{rel}} \|\boldsymbol{Y}\|_{\mathcal{S}_p}. \end{aligned}$$

Proposition A.6. *Fix $\eta_{\mathrm{abs}} \in [0, 1/2)$, $\eta_{\mathrm{rel}} \in [0, 1)$ and $k, q > 0$. Suppose that* SETH *holds. Then, there exist $C_d, C_B, \varepsilon > 0$ such that neither* $\mathsf{AbsPreBatchAtt}(n, m(n), d, B, \eta_{\mathrm{abs}})$ *nor* $\mathsf{RelPreBatchAtt}_p(n, m(n), d, B, \eta_{\mathrm{rel}})$ *can be solved with $\mathcal{O}(n^k)$ preprocessing time and $\mathcal{O}(m(n)n^{1-q})$ query time, for $d \coloneqq C \log n$, $B \coloneqq C_B\sqrt{\log n}$, any $p \in [1, \infty]$, and any function $m : \mathbb{N} \to \mathbb{N}$ with $m(n) \in [2n]$.*

Proof. Define shorthand notation $N \coloneqq 2n$ and $m \coloneqq m(n)$. Assume w.l.o.g. that $k > 1$, $q \in (0, 1/2)$, and that n is sufficiently large. We choose $C_d, \varepsilon > 0$ as the constant from Proposition A.3 so that $\mathsf{PreCP}(n, m, d, \varepsilon)$ cannot be solved with $\mathcal{O}(n^k)$ preprocessing time and $\mathcal{O}(mn^{1-q})$ query time for $d \coloneqq \frac{C_d}{2} \log n$. Suppose, for contradiction, that either $\mathsf{AbsPreBatchAtt}(n, m, D, B, \eta_{\mathrm{abs}})$ or $\mathsf{RelPreBatchAtt}_p(n, m, D, B, \eta_{\mathrm{rel}})$ can be solved with $\mathcal{O}(n^k)$ preprocessing time and $\mathcal{O}(mn^{1-q})$ decoding time.

We construct an algorithm for $\mathsf{PreCP}(n, m(n), d, \varepsilon)$ with $\mathcal{O}(n^k)$ preprocessing time and $\mathcal{O}(mn^{1-q})$ query time, contradicting Proposition A.3. To that end, let a threshold $t \in [0, d]$ and a set $\mathbb{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\} \subseteq \mathbb{M}_d$ be an input to $\mathsf{PreCP}(n, m, d, \varepsilon)$, together with queries $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m \in \mathbb{M}_d$ revealed after preprocessing. Applying Lemma A.4 to C_d and q gives a constant $c \in (0, C_d/2)$ so that we can decide the PreCP instance with $\mathcal{O}(nd) \leq \mathcal{O}(n^k)$ preprocessing time and $\mathcal{O}(mn^{1-q})$ decoding time if $t \leq c \log n$. We may there assume that $t > c \log n$. We may further assume that $(1+\varepsilon)t \leq d$. Indeed, if $(1+\varepsilon)t > d$, then every conceivable pair $(\boldsymbol{a}, \boldsymbol{b}) \in \{0,1\}^d \times \mathbb{B}$ has Hamming distance at most $d < (1+\varepsilon)t$, so the problem is trivial to decide.

We construct a KV cache $(\boldsymbol{K}, \boldsymbol{V})$ and a query $\boldsymbol{Q}$ such that the PreCP instance can be decided from an attention approximation after KV cache preprocessing. After seeing only t and $\mathbb{B}$, we apply Lemma A.5 with parameters ε, c and choose t as the threshold and $\mathbb{B}$ as the set. This supplies the constant $C_B > 0$ and allows us to construct, during preprocessing and in $\mathcal{O}(ND) = \mathcal{O}(n \log n) \leq \mathcal{O}(n^k)$ time, matrices $\boldsymbol{K} \in [-B, B]^{N \times D}$ and $\boldsymbol{V} \in [0,1]^{N \times D}$, which we preprocess with the hypothetical preprocessing algorithm for attention in $\mathcal{O}(n^k)$ time.

After that, let the vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m$ be revealed. Now, Lemma A.5 allows us to construct, in $\mathcal{O}(mD) = \mathcal{O}(m \log n) \leq \mathcal{O}(mn^{1-q})$ time, a matrix $\boldsymbol{Q} \in [-B, B]^{m \times D}$ such that $\boldsymbol{Y} \coloneqq \mathrm{Att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{y}\boldsymbol{e}_1^\top$ for some $\boldsymbol{y} \in \mathbb{R}^m$ with

$$y_i \geq \frac{n}{n+1} \qquad \text{if } \min_{j \in [n]} \|\boldsymbol{a}_i - \boldsymbol{b}_j\|_0 \leq t \tag{4}$$

and

$$y_i \leq \frac{1}{n+1} \qquad \text{if } \min_{j \in [n]} \|\boldsymbol{a}_i - \boldsymbol{b}_j\|_0 \geq (1+\varepsilon)t \tag{5}$$

for all $i \in [m]$. We use that to decide the PreCP instance. To that end, run the assumed approximate attention decoding algorithm on $\boldsymbol{Q}$ and let $\boldsymbol{T} \in \mathbb{R}^{m \times D}$ be its output.

Suppose first that the algorithm guarantees $\|\boldsymbol{T} - \boldsymbol{Y}\|_{\ell_\infty} \le \eta_{\text{abs}}$. If $\min_{i\in[m],j\in[n]} \|\boldsymbol{a}_i - \boldsymbol{b}_j\|_0 \le t$, then

$$T_{i,1} \ge Y_{i,1} - \eta_{\text{abs}} \overset{(4)}{\ge} \frac{n}{n+1} - \eta_{\text{abs}} > \frac{1}{2}$$

for some $i \in [m]$. If, instead, $\min_{i\in[m],j\in[n]} \|\boldsymbol{a}_i - \boldsymbol{b}_j\|_0 \ge (1+\varepsilon)t$, then

$$T_{i,1} \le Y_{i,1} + \eta_{\text{abs}} \overset{(5)}{\le} \frac{1}{n+1} + \eta_{\text{abs}} < \frac{1}{2}$$

for all $i \in [m]$. Thus, we can decide the PreCP instance by checking if any entry of $\boldsymbol{T}$ exceeds $1/2$.

Suppose now that, instead, the algorithm guarantees $\|\boldsymbol{T} - \boldsymbol{Y}\|_{\mathcal{S}_p} \le \eta_{\text{rel}}\|\boldsymbol{Y}\|_{\mathcal{S}_p}$. Since $\boldsymbol{Y} = \boldsymbol{y}\boldsymbol{e}_1^\top$, $\|\boldsymbol{Y}\|_{\mathcal{S}_p} = \|\boldsymbol{y}\|_2 \le \sqrt{m}$. If $\min_{i\in[m],j\in[n]} \|\boldsymbol{a}_i - \boldsymbol{b}_j\|_0 \le t$, then

$$\|\boldsymbol{T}\|_{\mathcal{S}_p} \ge \|\boldsymbol{Y}\|_{\mathcal{S}_p} - \|\boldsymbol{T} - \boldsymbol{Y}\|_{\mathcal{S}_p} \ge (1-\eta_{\text{rel}})\|\boldsymbol{Y}\|_{\mathcal{S}_p} = (1-\eta_{\text{rel}})\|\boldsymbol{y}\|_2 \overset{(4)}{\ge} (1-\eta_{\text{rel}})\frac{n}{n+1}.$$

for some $i \in [m]$. If, instead, $\min_{i\in[m],j\in[n]} \|\boldsymbol{a}_i - \boldsymbol{b}_j\|_0 \ge (1+\varepsilon)t$, then

$$\|\boldsymbol{T}\|_{\mathcal{S}_p} \le \|\boldsymbol{Y}\|_{\mathcal{S}_p} + \|\boldsymbol{T} - \boldsymbol{Y}\|_{\mathcal{S}_p} \le (1+\eta_{\text{rel}})\|\boldsymbol{Y}\|_{\mathcal{S}_p} = (1+\eta_{\text{rel}})\|\boldsymbol{y}\|_2 \overset{(5)}{\le} (1+\eta_{\text{rel}})\frac{\sqrt{m}}{n+1}.$$

Thus, we can decide the PreCP instance by checking if $\|\boldsymbol{T}\|_{\mathcal{S}_p} > (1-\eta_{\text{rel}})\frac{n}{n+1}$. Computing $\|\boldsymbol{T}\|_{\mathcal{S}_p}$ takes $\mathcal{O}(mD^2) = \mathcal{O}(m\log^2 n) \le \mathcal{O}(mn^{1-q})$ time. Either guarantee gives an algorithm for deciding the PreCP instance from $\boldsymbol{T}$ in $\mathcal{O}(D) = \mathcal{O}(\log n) \le \mathcal{O}(n^{1-q})$ time. Computing $\boldsymbol{T}$, by assumption, only requires $\mathcal{O}(n^k)$ preprocessing time and $\mathcal{O}(mn^{1-q})$ decoding time. That contradicts Proposition A.3 and thereby proves the claim. □

Theorem 3.7. *Fix $\eta_{\text{abs}} \in [0, \frac{1}{2})$, $\eta_{\text{rel}} \in [0,1)$, and $q > 0$. Assuming* SETH*, there exist $C_d, C_B > 0$ such that neither* $\mathsf{AbsAtt}(n, D, B, \eta_{\text{abs}})$ *nor* $\mathsf{RelAtt}_p(n, D, B, \eta_{\text{rel}})$ *can be solved in $\mathcal{O}(n^{2-q})$ time, for $D \coloneqq C_d \log n$, $B \coloneqq C_B\sqrt{\log n}$, and any $p \in [1,\infty]$.*

Proof. This follows directly from Proposition A.6 with $m(n) \coloneqq n$. □

Theorem 4.1. *Fix $\eta_{\text{abs}} \in [0, \frac{1}{2})$, $\eta_{\text{rel}} \in [0,1)$, and $k, q > 0$. Assuming* SETH*, there exist $C_d, C_B > 0$ such that neither* $\mathsf{AbsPreAtt}(n, D, B, \eta_{\text{abs}})$ *nor* $\mathsf{RelPreAtt}(n, D, B, \eta_{\text{rel}})$ *can be solved with $\mathcal{O}(n^k)$ preprocessing time and $\mathcal{O}(n^{1-q})$ decoding time, for $D \coloneqq C_d \log n$ and $B \coloneqq C_B\sqrt{\log n}$.*

Proof. This follows directly from Proposition A.6 with $m(n) \coloneqq 1$. □

A.4 Finding close pairs from a sparse attention approximation

We next show how a sparse attention algorithm can be used to identify approximate closest pairs. As before, we encode distances between a query vector and the preprocessed set as attention scores, with closer vectors receiving larger scores. We would then like to recover an approximate closest pair from any small set of keys carrying a constant fraction of the attention mass. For this argument to apply, however, the resulting attention instance must satisfy the sparsity promise of Problem 3. But a closest pair instance may contain many approximately closest vectors, causing the corresponding keys to receive similar scores and the attention mass to be spread among them. We overcome this by randomly perturbing the scores so that, with sufficiently high probability, the score corresponding to a close pair is amplified enough to dominate the attention. Exponential random variables provide exactly this effect. At the same time, the random perturbation must not allow a far pair to dominate by chance. We therefore truncate the exponential variables, limiting how much any score can be amplified while still allowing a close pair to separate from the rest. The following lemma shows that these truncated random variables indeed still create the required separation. Lemma A.8 then translates this separation into an attention instance in which, with high probability, an approximate closest pair carries almost all of the attention mass and can therefore be identified by a retrieval algorithm for sparse attention.

Lemma A.7. *Fix $L > 0$, $n \geq 2$, and $z_1, \dots, z_n \in \mathbb{R}$. Sample exponential random variables $E_1, \dots, E_n \overset{\text{iid}}{\sim} \text{Exp}(\theta)$ of rate $\theta > 0$ and clip them to $\widehat{E}_j \coloneqq \min\{E_j, L\}$. Then, for any $g > 0$,*

$$\Pr\Big[\exists j^\star \in [n] : \forall j^\star \neq j \in [n] : (s_{j^\star} + \widehat{E}_{j^\star}) - (s_j + \widehat{E}_j) \geq g\Big] \geq e^{-\theta g} - ne^{-\theta L}.$$

Proof. Define the scores $\widehat{Z}_j \coloneqq s_j + \widehat{E}_j$ and their corresponding values before clipping $Z_j \coloneqq s_j + E_j$. Fix an arbitrary $j^\star \in [n]$ and condition on $E_{-j^\star}$, that is, all E_j except $E_{j^\star}$. Among these, let

$$M_{j^\star} \coloneqq \max_{j \in [n],\, j \neq j^\star} Z_j$$

be the maximum score. Then,

$$\begin{aligned}\Pr\big[Z_{j^\star} \geq M_{j^\star} + g \mid E_{-j^\star}\big] &= e^{-\theta \max\{0, M_{j^\star} + g - s_{j^\star}\}} \\ &\geq e^{-\theta g} e^{-\theta \max\{0, M_{j^\star} - s_{j^\star}\}} = e^{-\theta g} \Pr\big[Z_{j^\star} \geq M_{j^\star} \mid E_{-j^\star}\big].\end{aligned}$$

Taking expectations over $E_{-j^\star}$,

$$\Pr\big[Z_{j^\star} \geq M_{j^\star} + g\big] \geq e^{-\theta g} \Pr\big[Z_{j^\star} \geq M_{j^\star}\big].$$

Summing over all possible choices of $j^\star$ and using the fact that the continuous random variables $s_j + E_j$ have a unique maximum almost surely,

$$\begin{aligned}\Pr\Big[\exists j^\star \in [n] : \forall j^\star \neq j \in [n] : Z_{j^\star} - Z_j \geq g\Big] &= \sum_{j^\star \in [n]} \Pr\big[Z_{j^\star} \geq M_{j^\star} + g\big] \\ &\geq e^{-\theta g} \sum_{j^\star \in [n]} \Pr\big[Z_{j^\star} \geq M_{j^\star}\big] \geq e^{-\theta g}.\end{aligned}$$

Finally, it may happen that some of the exponential random variables are clipped. Therefore, working on the even that no clipping occurs and using the union bound,

$$\begin{aligned}&\Pr\Big[\exists j^\star \in [n] : \forall j^\star \neq j \in [n] : \widehat{Z}_{j^\star} - \widehat{Z}_j \geq g\Big] \\ &\geq \Pr\Big[\exists j^\star \in [n] : \forall j^\star \neq j \in [n] : Z_{j^\star} - Z_j \geq g\Big] - \Pr\Big[\exists j \in [n] : E_j > L\Big] \\ &\geq e^{-\theta g} - ne^{-\theta L}.\end{aligned}$$

□

Lemma A.8. *Fix $\varepsilon, c, C_d > 0$ and $\gamma \in (0, 1)$. There exists $C_B \geq 0$ such that the following holds for all sufficiently large $n \in \mathbb{N}$, $d \coloneqq C_d \log n$, and $B \coloneqq C_B \sqrt{\log n}$.*

Choose $t \geq c \log n$ with $d \geq (1 + \varepsilon)t$, a set of vectors $\mathbb{B} = \{\boldsymbol{b}_1, \dots, \boldsymbol{b}_n\} \subseteq \mathbb{M}_d$, and a query $\boldsymbol{a} \in \mathbb{M}_d$. One can sample matrices $\boldsymbol{K} \in [-B, B]^{n \times 2d}$ independent of $\boldsymbol{a}$ and in $\mathcal{O}(nd)$ time, and construct a vector $\boldsymbol{q} \in [-B, B]^{2d}$ in $\mathcal{O}(d)$ time such that

$$\Pr\Big[\exists j^\star \in [n] : \|\boldsymbol{a} - \boldsymbol{b}_{j^\star}\|_0 \leq (1 + \varepsilon)t \;\; \textit{and} \;\; \text{softmax}_{j^\star}\big(\boldsymbol{K}\boldsymbol{q}/(2d)\big) > 1 - \frac{1}{n}\Big] \geq \frac{1}{2} n^{-\gamma}$$

if there exists $\boldsymbol{b}^\star \in \mathbb{B}$ with $\|\boldsymbol{a} - \boldsymbol{b}^\star\| \leq t$.

Proof. Define

$$\theta \coloneqq \frac{\gamma}{2}, \qquad R \coloneqq \frac{3}{\theta}, \qquad \text{and} \qquad \beta \coloneqq \frac{4d(R+1)\log n}{\varepsilon t}.$$

We first sample clipped exponential random variables

$$\widehat{E}_j \coloneqq \min\{E_j, R \log n\}, \qquad \text{where} \quad E_1, \dots, E_n \overset{\text{iid}}{\sim} \text{Exp}(\theta).$$

We then construct the key matrix and query vector as

$$\boldsymbol{K} \coloneqq \sqrt{\beta} \begin{bmatrix} \boldsymbol{b}_1^\top & \frac{2\widehat{E}_1}{\beta} \mathbf{1}_d^\top \\ \vdots & \vdots \\ \boldsymbol{b}_n^\top & \frac{2\widehat{E}_n}{\beta} \mathbf{1}_d^\top \end{bmatrix} \in \mathbb{R}^{n \times 2d} \qquad \text{and} \qquad \boldsymbol{q} \coloneqq \sqrt{\beta} \begin{bmatrix} \boldsymbol{a}^\top & \mathbf{1}_d^\top \end{bmatrix} \in \mathbb{R}^{2d}.$$

Next, choose C_B so that the constructed matrices satisfy the required entry bounds. Since $d = C_d \log n$ and $t \geq c \log n$,

$$\beta = \frac{4d(R+1)\log n}{\varepsilon t} \leq \frac{4C_d(R+1)\log n}{\varepsilon c}.$$

Moreover, since $t \leq d/(1+\varepsilon)$ and $\widehat{E}_j \leq R \log n$,

$$0 \leq \frac{2\widehat{E}_j}{\beta} \leq \frac{R\varepsilon}{2(R+1)(1+\varepsilon)} < \frac{1}{2}.$$

Consequently, there is a constant C_B depending only on C_d, ε, c, and R such that $\|\boldsymbol{K}\|_{\ell_\infty} \leq C_B\sqrt{\log n}$ and $\|\boldsymbol{q}\|_{\ell_\infty} \leq C_B\sqrt{\log n}$.

By construction, the logit between the query and key $j \in [n]$ is

$$\begin{aligned}
\widehat{Z}_j &\coloneqq \frac{\boldsymbol{K}_j^\top \boldsymbol{q}}{2d} = \frac{\beta}{2d}\Big(\langle \boldsymbol{a}, \boldsymbol{b}_j\rangle + \frac{2d\widehat{E}_j}{\beta}\Big) = \frac{\beta}{2d}\langle \boldsymbol{a}, \boldsymbol{b}_j\rangle + \widehat{E}_j \\
&= \frac{\beta}{4d}\big(\|\boldsymbol{a}\|_2^2 + \|\boldsymbol{b}_j\|_2^2 - \|\boldsymbol{a}-\boldsymbol{b}_j\|_0\big) + \widehat{E}_j = \frac{\beta}{4d}\big(d - \|\boldsymbol{a}-\boldsymbol{b}_j\|_0\big) + \widehat{E}_j \\
&= \frac{\beta}{4}\Big(1 - \frac{\|\boldsymbol{a}-\boldsymbol{b}_j\|_0}{d}\Big) + \widehat{E}_j, \qquad (6)
\end{aligned}$$

where we use that $\|\boldsymbol{a}\|_0 = \|\boldsymbol{b}\|_0 = d/2$. Suppose now there is some close partner for $\boldsymbol{a}$ at index $i \in [n]$ with $\|\boldsymbol{a}-\boldsymbol{b}_i\|_0 \leq t$. Then, for all far vectors at index $j \in [n]$ with $\|\boldsymbol{a}-\boldsymbol{b}_j\|_0 > (1+\varepsilon)t$,

$$\begin{aligned}
\widehat{Z}_i - \widehat{Z}_j &\overset{(6)}{=} \frac{\beta}{4d}\big(\|\boldsymbol{a}-\boldsymbol{b}_j\|_0 - \|\boldsymbol{a}-\boldsymbol{b}_i\|_0\big) + \widehat{E}_i - \widehat{E}_j \\
&\geq \frac{\beta\varepsilon t}{4d} + \widehat{E}_i - \widehat{E}_j \\
&\geq (R+1)\log n - R\log n > 0,
\end{aligned}$$

so logits of close partners are separated from those of far partners. In particular,

$$j^\star \coloneqq \operatorname*{arg\,max}_{j\in[n]} \widehat{Z}_j \in \big\{j \in [n] : \|\boldsymbol{a}-\boldsymbol{b}_j\|_0 \leq (1+\varepsilon)t\big\}. \qquad (7)$$

Now, compare this maximal logit at position $j^\star$ against the other logits. Applying Lemma A.7 with $\theta = \gamma/2$, $L = R\log n$, and $g = 2\log n$ gives

$$\Pr\Big[\forall j^\star \neq j \in [n] : \widehat{Z}_{j^\star} - \widehat{Z}_j \geq 2\log n\Big] \geq n^{-\gamma} - n^{-2} \geq \frac{1}{2}n^{-\gamma}$$

for all sufficiently large n. On the same event,

$$1 - \mathrm{softmax}_{j^\star}(\boldsymbol{q}\boldsymbol{K}^\top/(2d)) = \frac{\sum_{j^\star\neq j\in[n]} e^{\widehat{Z}_j}}{e^{\widehat{Z}_j} + \sum_{j^\star\neq j\in[n]} e^{\widehat{Z}_j}} \leq \frac{(n-1)e^{\widehat{Z}_j - 2\log n}}{e^{\widehat{Z}_j}} = \frac{n-1}{n^2} \leq \frac{1}{n},$$

which proves the claim. □

A.5 Proof of Theorem 5.1

We can now prove Theorem 5.1 by applying Lemma A.8 to the closest pair problem from Definition A.2. The construction identifies an approximate closest pair by letting its key receive almost all the attention for a given query, with some probability decreasing polynomially in the input size n. To ensure a large success probability, we construct and preprocess sufficiently many independent copies, which increases preprocessing only polynomially and can be chosen so that the total decoding time remains truly sublinear. On a successful copy, every valid sparse-attention output must contain the dominant key.

Theorem 5.1. *Fix any $\alpha \in (0,1)$ and $k, q > 0$. Assuming* SETH*, there exist $C_d, C_B > 0$ such that* $\mathsf{SparseAtt}(n, D, B, \alpha)$ *cannot be solved with $\mathcal{O}(n^k)$ preprocessing time and $\mathcal{O}(n^{1-q})$ decoding time, for $D \coloneqq C_d \log n$ and $B \coloneqq C_B\sqrt{\log n}$.*

Proof. Assume w.l.o.g. that $k > 1$ and $q \in (0, 1/2)$. We choose $C_d, \varepsilon > 0$ as the constant from Proposition A.3 so that $\mathsf{PreCP}(n, 1, d, \varepsilon)$ cannot be solved with $\mathcal{O}(n^{k+1})$ preprocessing time and $\mathcal{O}(n^{1-q/2})$ decoding time for $d \coloneqq \frac{C_d}{2} \log n$. Suppose, for contradiction, that $\mathsf{SparseAtt}(n, D, B, \alpha)$ can be solved with $\mathcal{O}(n^k)$ preprocessing time and $\mathcal{O}(n^{1-q})$ decoding time.

We construct an algorithm for $\mathsf{PreCP}(n, 1, d, \varepsilon)$ with $\mathcal{O}(n^{k+1})$ preprocessing time and $\mathcal{O}(n^{1-q/2})$ query time, contradicting Proposition A.3. To that end, let a threshold $t \in [0, d]$ and a set $\mathbb{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\} \subseteq \mathbb{M}_d$ be an input to $\mathsf{PreCP}(n, 1, d, \varepsilon)$, together with a query $\boldsymbol{a} \in \mathbb{M}_d$ revealed after preprocessing. Applying Lemma A.4 to C_d and q gives a constant $c \in (0, C_d/2)$ so that we can decide the PreCP instance with $\mathcal{O}(nd) \leq \mathcal{O}(n^k)$ preprocessing time and $\mathcal{O}(n^{1-q})$ decoding time if $t \leq c \log n$. We may there assume that $t > c \log n$. We may further assume that $(1 + \varepsilon)t \leq d$. Indeed, if $(1 + \varepsilon)t > d$, then every conceivable pair $(\boldsymbol{a}, \boldsymbol{b}) \in \{0, 1\}^d \times \mathbb{B}$ has Hamming distance at most $d < (1 + \varepsilon)t$, so the problem is trivial to decide.

We show that the PreCP instance can be decided by finding a small set of keys in $\boldsymbol{K}$ receiving at least an α-fraction of the attention for the query $\boldsymbol{q}$. After seeing only t and $\mathbb{B}$, we apply Lemma A.8 with parameters ε, c, C_d, and $\gamma = q/4$. This supplies the constant $C_B > 0$ and allows us to sample, during preprocessing and in $\mathcal{O}(Tnd) \leq \mathcal{O}(n^{k+1})$ time, $T \coloneqq C_T n^\gamma$ independent key matrices $\boldsymbol{K}^{(1)}, \ldots, \boldsymbol{K}^{(T)} \in [-B, B]^{n \times D}$, which we preprocess with the hypothetical preprocessing algorithm for sparse attention in $\mathcal{O}(Tn^k) \leq \mathcal{O}(n^{k+1})$ time.

After that, let the vector $\boldsymbol{a} \in \mathbb{M}_d$ be revealed. Now, Lemma A.8 allows us to construct, in $\mathcal{O}(Td) \leq \mathcal{O}(n^{1-q})$ time, the queries $\boldsymbol{q}^{(1)}, \ldots, \boldsymbol{q}^{(T)} \in [-B, B]^D$ corresponding to the key matrices and guaranteeing that

$$\Pr\Big[\exists j^\star \in [n] : \|\boldsymbol{a} - \boldsymbol{b}_{j^\star}\|_0 \leq (1 + \varepsilon)t \text{ and } \operatorname{softmax}_{j^\star}\big(\boldsymbol{K}^{(\ell)}\boldsymbol{q}^{(\ell)}/D\big) > 1 - \frac{1}{n}\Big] \geq \frac{1}{2}n^{-\gamma} \quad (8)$$

for each copy $\ell \in [T]$ if $\min_{j \in [n]} \|\boldsymbol{a} - \boldsymbol{b}_j\|_0 \leq t$. Again for each copy, we run the hypothetical decoding algorithm for sparse attention, resulting in explicit sets $\mathbb{I}^{(1)}, \ldots, \mathbb{I}^{(T)} \subseteq [n]$. Each invocation runs in $\mathcal{O}(n^{1-q})$ time by assumption, so, in particular, $|\mathbb{I}^{(\ell)}| = \mathcal{O}(n^{1-q})$ for each $\ell \in [T]$. For each copy $\ell \in [T]$ and each index $i \in \mathbb{I}^{(\ell)}$, we check if $\|\boldsymbol{a} - \boldsymbol{b}_i\|_0 \leq (1 + \varepsilon)t$. If we find such an index i, we accept the PreCP instance. Otherwise, reject the instance.

The algorithm never falsely accepts an instance. If, however, $\min_{j \in [n]} \|\boldsymbol{a} - \boldsymbol{b}_j\|_0 \leq t$, then it follows from (8) that only with probability

$$\Big(1 - \frac{1}{2}n^{-\gamma}\Big)^{C_T n^\gamma} \leq e^{-C_T/2}$$

no copy contains an index receiving more than $1 - \frac{1}{n}$ attention. Conversely, we can choose C_T as a sufficiently large absolute constant so that, with probability at least $2/3$, some copy $\ell \in [T]$ has an index $i \in [n]$ with

$$\operatorname{softmax}_i\big(\boldsymbol{K}^{(\ell)}\boldsymbol{q}^{(\ell)}/D\big) > 1 - \frac{1}{n}.$$

On such a copy, assuming n is sufficiently large, $\mathbb{I}^{(\ell)}$ can only receive at least α attention if it includes the index i. Consequently, the hypothetical decoding algorithm for sparse attention guarantees $i \in \mathbb{I}^{(\ell)}$, so our algorithm for PreCP accepts.

Put together, we obtain a bounded-error algorithm for $\mathsf{PreCP}(n, 1, d, \varepsilon)$. The algorithm uses $\mathcal{O}(n^{k+1})$ time during preprocessing and $\mathcal{O}(n^\gamma n^{1-q} d) \leq \mathcal{O}(n^{1-q/2})$ time during decoding. That contradicts Proposition A.3 and thereby proves the claim. □