SOCKET: SOFT COLLISION KERNEL ESTIMATOR FOR SPARSE ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Exploiting sparsity is key to efficient long-context inference, as attention dominates the cost of autoregressive decoding. Sparse attention reduces this cost by restricting computation to a subset of tokens, but its effectiveness hinges on fast and accurate token scoring and selection at inference time. Data-agnostic approaches offer an attractive way to perform this selection, but often incur substantial memory overhead to maintain high recall. We revisit Locality-Sensitive Hashing (LSH) and introduce SOCKET, a SOft Collision Kernel EsTimator that replaces hard bucket matches with probabilistic, similarity-aware aggregation. Traditional LSH relies on binary collision signals, providing limited information for ranking tokens and necessitating many hash tables for accurate retrieval. In contrast, soft LSH accumulates graded collision evidence across hash tables, closely preserving the true top-k ordering with significantly less memory. This reframes LSH from a candidate-generation mechanism into a principled scoring kernel for sparse attention. Building on this insight, SOCKET enables efficient token selection without ad hoc voting and matches or outperforms existing sparse attention methods across multiple long-context benchmarks and diverse language models. With a custom set of CUDA/Triton kernels for scoring, selection, and attention, SOCKET achieves up to approximately $1.5\times$ higher throughput than FlashAttention.

1 INTRODUCTION

Large language models (LLMs), such as GPT (Brown et al., 2020) and Llama (Touvron et al., 2023), have achieved remarkable performance in next-token prediction, enabling a wide range of applications in language understanding and generation (Achiam et al., 2023; Chowdhery et al., 2023; Hoffmann et al., 2022; Raffel et al., 2020). To support more powerful in-context learning and increasingly complex applications (Dong et al., 2024; Press et al., 2022; Ratner et al., 2023; Wei et al., 2022), the maximum input length supported during inference has grown rapidly–from 2K-4K tokens (Devlin et al., 2019; Vaswani et al., 2017) to 32K (Beltagy et al., 2020; Chowdhery et al., 2023), 128K (Achiam et al., 2023; Touvron et al., 2023), and even millions of tokens (Liu et al., 2024). Beyond language, LLMs have also been extended to multimodal domains such as vision, video, code, databases, and scientific discovery (Jumper et al., 2021; Li et al., 2023a;b; Radford et al., 2021; Ruan & Jin, 2022; Zhou et al., 2024).

LLM inference consists of an initial prefilling phase followed by iterative decoding steps that generate one token at a time. During prefilling, the model processes the full input sequence and computes key-value (KV) representations for all tokens. During decoding, only the most recently generated token is processed, and its KV representations are appended to the cache, avoiding redundant computation. However, each decoding step requires the new token to attend to all previously cached tokens, making attention computation increasingly memory-bound as the context grows. When the KV cache exceeds GPU memory capacity and is partially offloaded to CPU memory, additional CPU-GPU transfers further exacerbate this bottleneck. These effects reveal a fundamental scalability limitation of dense attention for long-context inference. Sparse attention addresses this limitation by restricting attention to a selected subset of tokens, reducing memory movement while approximating dense attention behavior.

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.

Method	nm2	nm3	vt	fwe[1]	qa1	qa2	Avg
Ministral-3-8B-Instruct-2512							
20× sparsity							
Quest (ps=16)	95	65	99.2	97.6	78	58	82.1
DS (l=16, ch=32)	90	39	98	90	73	53	73.8
PQCache (pq=8, gr=4)	99	97	100	91.3	81	60	88.1
MagicPig (P=8, L=75)	2.3	0	98.3	90.5	44.6	38.6	45.7
SOCKET (P=10, L=60)	**99**	93	**100**	**98**	80	**60**	**88.3**
50× sparsity							
Quest (ps=8)	91	64	100	95.6	79	57	81.1
DS (l=16, ch=32)	63	4	88.4	85.6	60	49	58.3
PQCache (pq=8, gr=4)	95	82.6	100	89.8	81.6	56.6	84.3
MagicPig (P=10, L=150)	0.3	0	93.3	90.5	34.3	31	41.6
SOCKET (P=10, L=60)	**98**	82	**100**	91.1	79.3	57.6	**84.7**
Llama-3.1-8B-Instruct							
20× sparsity							
Quest (ps=16)	89	69	95	90	81	53	79.5
DS (l=16, ch=32)	49	82	90	91	68	51	71.8
HashAttn (b=128)	73	56	82	90	76	49	71.0
PQCache (pq=8, gr=4)	93	94	89.2	81.0	81	50	81.4
MagicPig (P=8, L=75)	1	0	75.4	84.8	31	33	37.5
SOCKET (P=10, L=60)	**93**	**94**	**91.4**	**94.7**	80	**51.3**	**84.1**
50× sparsity							
Quest (ps=16)	52	7	91	88	65	51	59.0
DS (l=16, ch=32)	28	8	84	76	48	44	48.0
HashAttn (b=128)	40	7	74	86	64	45	52.7
PQCache (pq=8, gr=4)	72	79	89.0	81.0	77	50	74.7
MagicPig (P=10, L=150)	0.3	0	70.6	82.7	21.6	28	33.9
SOCKET (P=10, L=60)	**80**	73	83.8	**90.3**	**77**	**50**	**75.7**

Table 1: Performance on RULER-HARD-32K using Ministral-3-8B and Llama-3.1-8B.

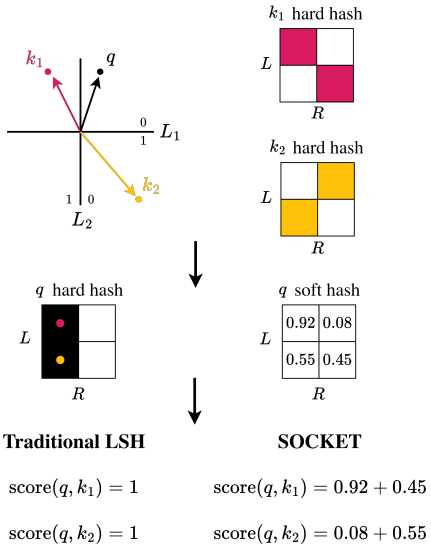


Figure 1: Traditional LSH assigns binary scores based on hash collisions, while soft LSH produces continuous scores via probabilistic bucket assignments. Soft LSH therefore induces a stable ranking: if k_1 is closer to q than k_2, then $\text{score}(q, k_1) > \text{score}(q, k_2)$.

The attention output for a query $\mathbf{q}$ under scaled dot-product attention (SDPA) is given by

$$\mathbf{y}(\mathbf{q}) = \sum_{i=1}^{n} a_i \mathbf{v}_i, \tag{1}$$

where $a_i = \exp(\mathbf{k}_i^\top \mathbf{q}) / \sum_{j=1}^{n} \exp(\mathbf{k}_j^\top \mathbf{q})$ and, $\mathbf{k}_i, \mathbf{v}_i \in \mathbb{R}^d$ denote the key and value vectors associated with token i, and a_i represents its attention weight (for simplicity, we remove the $1/\sqrt{d}$ factor in the definition). The central challenge in approximating dense attention lies in identifying the tokens that contribute most to this sum. Prior work has shown that, in hindsight, the dominant contributors are those with large values of $a_i \|\mathbf{v}_i\|_2$ (Desai et al., 2025). This observation motivates selecting tokens with the largest query–key inner products $\mathbf{k}_i^\top \mathbf{q}$, commonly referred to as *top-k* selection. Related extensions, such as top-p, further adapt this strategy by dynamically allocating token budgets across attention heads. Consequently, much of the sparse attention literature has focused on efficiently approximating top-k selection (Desai et al., 2025; Gupta et al., 2021; Hooper et al., 2025; Tang et al., 2024; Yang et al., 2024; Zhang et al., 2025) or top-p selection (Zhu et al., 2026).

Locality-Sensitive Hashing (LSH) (Indyk & Motwani, 1998), in particular, is a widely used randomized technique for identifying similar keys in high-dimensional spaces. In its traditional form, LSH applies random projections followed by a sign function, producing binary collision indicators between a query and a key. However, this *hard* formulation is poorly suited for ranking stability of important keys: binary collisions cannot express partial similarity, leading to coarse and noisy rankings (see fig. 2). Soft LSH resolves this limitation by aggregating collision evidence across multiple hash tables to produce lightweight, similarity-aware scores for every key. Recently, randomized data structures have emerged as a promising approach for approximating attention. RACE-based sketches (Coleman et al., 2020; Coleman & Shrivastava, 2020) have been used to approximate softmax attention in linear time (Joshi et al., 2026), while LSH-based methods have been explored to approximate attention and retrieval (Gionis et al., 1999; Kitaev et al., 2020; Liu et al., 2011). These works highlight the potential of randomized, data-agnostic primitives to reduce attention cost without relying on expensive data-dependent preprocessing.

[1]For fwe, SOCKET uses $(P = 10, L = 10, \tau = 1e^{-3})$

Unlike most existing sparse attention methods, SOCKET is *data-agnostic*. This design choice yields three key advantages. First, it enables substantially faster Time-To-First-Token (TTFT) compared to data-dependent clustering approaches such as k-means, as shown in fig. 3a. Second, it enables deployment without retraining or calibration, making it robust to distribution shifts during inference. Finally, it allows for interpretable theoretical guarantees with respect to a kernel that closely mimics the softmax kernel, as formalized in Theorem 3.

Table 2: Computational cost and memory overhead during retrieval for SOCKET and traditional LSH. All results are benchmarked using our custom CUDA scoring kernel. Avg Score represents the average retrieval accuracy for RULER-HARD-32K dataset using Llama-3.1-8B.

Method	(P, L)	Memory (GB)	Overhead	Time (ms)	Overhead	Avg Score
SOCKET	(10, 60)	1.048	1.00×	0.387	1.00×	**85.08**
LSH	(10, 60)	1.049	1.00×	0.387	1.00×	10.00
LSH	(2, 300)	2.953	2.81×	1.027	2.65×	84
LSH	(2, 400)	3.753	3.57×	1.328	3.43×	83.76
LSH	(2, 500)	4.554	4.34×	1.629	4.20×	84.80

Key Idea: Our preliminary experiments show that *hard* LSH fails to rank candidate keys by their true relevance under the same memory budget (fig. 2), motivating a key question: *Can LSH be augmented to better rank candidates by relevance?*

In SOCKET, each key is assigned a continuous score based on soft collision probabilities across multiple hash tables, weighted by the corresponding value vector norms. This turns LSH from a binary filter into a similarity-aware scoring kernel that enables stable key ranking. Crucially, these scores allow efficient key selection without accessing full key or value vectors. From a systems perspective, exact top-k selection requires reading full key vectors (e.g., 128 bfloat values per token), whereas SOCKET uses compact hash representations, retrieving only precomputed bucket indices (e.g., $\sim$600 bits per token) to compute query-dependent soft counts, along with a single integer (value norm) per key. This substantially reduces memory traffic during decoding while preserving ranking fidelity. We present ablations of both SOCKET and a *hard* LSH estimator in Sections A.1 and A.2. The results show that traditional LSH requires many more hash tables to match SOCKET's retrieval accuracy, incurring higher memory overhead and latency (Table 2). Overall, SOCKET offers a better accuracy-efficiency trade-off for data-agnostic sparse attention. In addition to these findings and extensive empirical validation, we make the following contributions:

I. Soft LSH as a *stable ranker*: We propose a *data-agnostic soft* LSH mechanism that serves as a substantially more effective ranking function than LSH, making it well suited for sparse attention.

II. Efficiency with SOCKET: We design a custom CUDA kernel for scoring of keys and show a $1.5\times$ throughput speedup over FlashAttention during decoding using GPT-FAST[1].

III. Theoretical Insights: We provide theoretical guarantees showing that soft LSH closely approximates a softmax-like attention kernel. Additionally, we theoretically show why soft LSH yields more accurate and stable token rankings than traditional LSH.

2 Related Work

Given the importance of long-context inference, extensive prior work accelerates attention by approximating top-k selection and restricting computation to a subset of tokens per query. One line of work uses dimensionality reduction: Double Sparsity (Yang et al., 2024) reduces computation along the feature dimension by selecting channels via offline calibration of channel norms, while Quest (Tang et al., 2024) reduces computation along the token dimension by selecting tokens at the page level. Although effective, these approaches rely on data-dependent heuristics and lack theoretical guarantees. In Section 6, we show that SOCKET matches or outperforms these methods on diverse set of models while providing a principled, data-agnostic alternative. A second line of work casts top-k attention as retrieval. RetrievalAttention (Liu et al., 2025) uses graph-based nearest neighbor search but offloads selection to the CPU due to irregular computation, introducing latency. PQCache (Zhang et al., 2025) integrates product quantization into KV-cache management for approximate retrieval, also relying on a CPU–GPU pipeline. In contrast, we show that

[1] https://github.com/meta-pytorch/gpt-fast

data-agnostic random projections significantly accelerate TTFT while maintaining or improving accuracy (fig. 3a). SOCKET further macthes or outperforms PQCACHE on a diverse set of models, while offering a simpler design (see Section 6).

Furthermore, several approaches estimate attention outputs using hashing-based data structures. LEARNING-TO-HASH ATTENTION (Sun et al., 2022) trains hash functions to map keys into balanced hash tables, enabling sparse retrieval during attention computation. HASHATTENTION (Desai et al., 2025) encodes queries and keys into Hamming space using learned mappings to capture semantic similarity. MAGICPIG (Chen et al., 2025) addresses the bias of top-k attention under relatively uniform attention distributions using an LSH-based importance-sampling estimator. Specifically, it samples candidate keys via traditional LSH and applies importance-sampling correction to obtain an unbiased estimate of softmax attention, requiring substantial memory for accurate sampling. In contrast, SOCKET uses *soft* LSH to score and deterministically retrieve top-k keys, followed by exact attention over the selected subset. Thus, while MAGICPIG is a sampling-based estimator, SOCKET directly targets accurate key ranking with substantially lower memory overhead. In Section 6, we show that SOCKET improves performance on aggregation, summarization, and retrieval tasks while using significantly less memory than MAGICPIG.

Several other works address efficient long-context inference, including VATTENTION (Desai et al., 2026), SQUEEZE ATTENTION (Hooper et al., 2025), LOKI (Singhania et al., 2024), INFLLM (Xiao et al., 2024a), STREAMLLM (Xiao et al., 2024b), ADAMAS (Yan et al., 2025), and H_2O (Zhang et al., 2023), which explore complementary system-level and algorithm-level strategies to reduce attention overhead. We do not aim to provide a comprehensive empirical comparison with these methods. Instead, we focus on a principled, data-agnostic sparsification approach based on locality-sensitive hashing, which serves as a similarity-aware scoring mechanism rather than a binary filtering heuristic. Accordingly, we do not discuss these methods further.

3 BACKGROUND

3.1 LOCALITY-SENSITIVE HASHING (LSH)

An LSH family $\mathcal{H}$ for a similarity Sim makes near pairs collide more often than far pairs. Formally, $\mathcal{H}$ is (S_0, cS_0, p_1, p_2)-sensitive if for all $x, y \in \mathbb{R}^D$,

$$\begin{cases} \mathrm{Sim}(x,y) \geq S_0 \;\Rightarrow\; \Pr_{h\sim\mathcal{H}}[h(x) = h(y)] \geq p_1, \\ \mathrm{Sim}(x,y) \leq cS_0 \;\Rightarrow\; \Pr_{h\sim\mathcal{H}}[h(x) = h(y)] \leq p_2, \end{cases}$$

where $p_1 > p_2$ and $c < 1$. Such families enable sublinear-time approximate nearest-neighbor data structures. A convenient sufficient condition, satisfied by SimHash and WTA (Charikar, 2002; Chen & Shrivastava, 2018; Yagnik et al., 2011), is that the collision probability is a monotone function of similarity, $\Pr_{h\sim\mathcal{H}}[h(x) = h(y)] = f(\mathrm{Sim}(x,y))$ with f increasing.

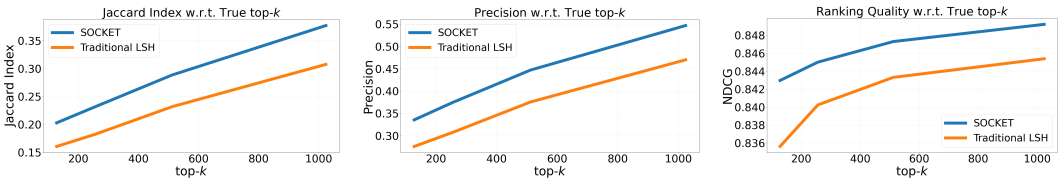
3.2 SPARSE ATTENTION

For clarity, we restrict our exposition to the case of batch size one with a single query vector $\mathbf{q} \in \mathbb{R}d$. Consider keys $\mathbf{k}_1, \ldots, \mathbf{k}_N \in \mathbb{R}^d$ and corresponding values $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N \in \mathbb{R}^d$. We modify equation 1 to define sparse attention. Let S denote the sequence of k important token indices selected by a method. The sparse attention computation based on this index set is given by:

$$\mathbf{y}_k(\mathbf{q}) = \sum_{i\in S_k} \alpha_i \mathbf{v}_i\,, \quad \text{where } \alpha_i = \frac{\exp(\mathbf{k}_i^\top \mathbf{q})}{\sum_{j\in\mathcal{S}_k} \exp(\mathbf{k}_j^\top \mathbf{q})} \;\text{ for } i \in \mathcal{S}. \tag{2}$$

4 INTRODUCING SOCKET

In dense attention, a query vector $\mathbf{q}$ interacts with all N keys to produce attention weights, which are then applied to the corresponding values to generate the output embedding. Prior work has shown that, for token generation, a small subset of keys often dominates the attention mass (Ge et al., 2024;



Jaccard Precision NDCG

Figure 2: Ranking quality comparison between SOCKET and traditional LSH under same memory budget (600 bits/token). Keys and queries are extracted from the final layer of Llama-3.1-8B on the Qasper dataset, and the ground-truth relevance is defined by the dot-product similarity between each key–query pair. Precision, Jaccard Index, and NDCG are well defined in the Appendix A.5.

Algorithm 1 PrecomputeKeyHashes (prefill)

Input: $\{\mathbf{k}_j, \boldsymbol{b}_j\}_{j=1}^{N}$, #planes P, #tables L
$R \leftarrow 2^P$
for $\ell = 1$ to L **do**
  Sample $\mathbf{W}^{(\ell)} \sim \mathcal{N}(0, 1)$
  for $j = 1$ to N **do**
    $h^{(\ell)}(\mathbf{k}_j) \leftarrow \text{sign}(\mathbf{W}^{(\ell)}\mathbf{k}_j)$
    Encode $h^{(\ell)}(\mathbf{k}_j)$ as BucketId $b_j^{(\ell)}$
  end for
end for
return $\{||\boldsymbol{b}_j||_2\}, \{\mathbf{W}^{(\ell)}\}, \{b_j^{(\ell)}\}$

Table 3: Correlation and variance comparison on SAMSUM and QASPER.

		SAMSUM		QASPER	
P	L	Corr	Var	Corr	Var
SOCKET ($\tau = 0.5$)					
10	20	0.504	8.9×10^{-9}	0.483	7.7×10^{-9}
10	40	0.625	4.4×10^{-9}	0.606	4.6×10^{-9}
10	60	0.690	3.03×10^{-9}	0.673	2.7×10^{-9}
Hard LSH					
2	250	0.573	5×10^{-4}	0.550	5×10^{-4}
2	300	0.605	4×10^{-4}	0.583	4×10^{-4}
2	350	0.633	3×10^{-4}	0.610	3×10^{-4}

Zhang et al., 2023). We leverage this observation by using soft LSH as a sparsification mechanism, treating it as a scoring function to identify the most relevant keys.

In traditional LSH-based retrieval, each key $\mathbf{k}_j$ is projected into L independent hash tables. For each table ℓ, the key is assigned to a discrete bucket $b_j^{(\ell)}$. At inference time, traditional LSH hashes the query $\mathbf{q}$ in the same manner and scores a key by counting the number of hash tables in which it collides with the query. We denote this hard collision score by $s_{\text{hard}}(\mathbf{k}_j, \mathbf{q})$. In contrast, SOCKET replaces the hard query assignment with a soft assignment: while keys are still assigned to a single bucket per hash table, the query distributes probability mass across all buckets. We denote the resulting SOCKET soft collision score by $s_{\text{soft}}(\mathbf{k}_j, \mathbf{q})$. These two scores are defined as

$$s_{\text{hard}}(\mathbf{k}_j, \mathbf{q}) = \sum_{\ell=1}^{L} \mathbb{I}\left[b_j^{(\ell)} = b_{\mathbf{q}}^{(\ell)}\right], \qquad s_{\text{soft}}(\mathbf{k}_j, \mathbf{q}) = \sum_{\ell=1}^{L} p_\tau^{(\ell)}\left(b_j^{(\ell)} \mid \mathbf{q}\right). \tag{3}$$

SOCKET assigns higher scores to keys whose hash buckets receive greater probability mass under the query's soft hash. We use these scores to select the top-k keys, on which attention is subsequently computed. A clean schematic that distinguishes traditional LSH and soft LSH as a stable ranker can be seen in fig. 1.

4.1 THE FINAL ALGORITHM

Our method consists of three stages:

Algorithm 1 (Keys hashing): During the prefill phase, each key vector is hashed into L independent hash tables. Specifically, we apply L random projections and assign each key to a single bucket per hash table. These bucket assignments are computed once and cached in GPU memory for reuse during decoding, together with the corresponding value vector norms.
Algorithm 2 (Query soft hashing): At decoding time, a new query vector is softly hashed into the same L hash tables. Instead of a single bucket assignment, the query induces a probability distribution over buckets in each table. Using these distributions, we compute a score for every

Algorithm 2 SoftBucketProbs (decoding)

Input: $\mathbf{q}$, $\{\mathbf{W}^{(\ell)}\}$, $\{\mathbf{c}_r\}_{r=1}^{R} \subset \{\pm 1\}^P$, τ
for $\ell = 1$ to L **do**
  $\mathbf{u}^{(\ell)}(\mathbf{q}) \leftarrow \frac{1}{\sqrt{d}} \tanh(\mathbf{W}^{(\ell)}\mathbf{q}) \in \mathbb{R}^P$
  for $r = 1$ to R **do**
    $\text{logit}^{(\ell)}(r) \leftarrow \frac{\mathbf{u}^{(\ell)}(\mathbf{q})^\top \mathbf{c}_r}{\tau}$
  end for
  $p_\tau^{(\ell)}(\cdot \mid \mathbf{q}) \leftarrow \text{softmax}(\text{logits})$
end for
return $p_\tau^{(\ell)}(r \mid \mathbf{q})\ \forall\, r \in [R]$ and $\ell \in [L]$.

Algorithm 3 ValueAwareTop-k Attention

Input: $\mathbf{q}$, $p_\tau^{(\ell)}$, $b_j^{(\ell)}$, values $\boldsymbol{b}_j$, L, k
for $j = 1$ to N **do**
  $\widehat{w}_j \leftarrow \sum_{\ell=1}^{L} p_\tau^{(\ell)}(b_j^{(\ell)} \mid \mathbf{q})$
end for
$\mathcal{S}_k \leftarrow \text{TopK}(\widehat{w}_1\|\boldsymbol{b}_1\|_2, \ldots, \widehat{w}_N\|\boldsymbol{b}_N\|_2)$
$\alpha_j \leftarrow \frac{\exp(\mathbf{k}_j^\top \mathbf{q})}{\sum_{i \in \mathcal{S}_k} \exp(\mathbf{k}_i^\top \mathbf{q})}$ for all $j \in \mathcal{S}_k$
return $\mathbf{y}_k(\mathbf{q}) = \sum_{j \in \mathcal{S}_k} \alpha_j \boldsymbol{b}_j$

key by aggregating the probability mass assigned to the key's corresponding buckets, as defined in equation 3.

Algorithm 3 (Top-k selection): Finally, we select the top-k keys according to their soft LSH scores and compute attention using only this subset. This yields a sparse yet high-quality approximation of dense attention. We discuss approximation quality in the next section.

5 Theoretical Insights

In this section, we present the theoretical analysis of our method. We characterize the quality of the similarity scores produced by soft LSH and study a sampling-based estimator to establish concentration and approximation guarantees with respect to cosine similarity–based (angular) attention. We adopt angular attention as a modeling choice, as it provides an interpretable and analytically tractable surrogate for the attention scoring function. Concretely, for a fixed query $\mathbf{q}$ and keys $\{\mathbf{k}_j\}_{j=1}^{N}$, we define the angular kernel weights

$$w_j \;:=\; \left(1 - \frac{1}{\pi}\cos^{-1}\!\left(\frac{\mathbf{q}^\top \mathbf{k}_j}{\|\mathbf{q}\|\,\|\mathbf{k}_j\|}\right)\right)^P \in [0,1], \tag{4}$$

where P controls the sharpness of the kernel. Normalizing these weights yields the angular attention distribution and the corresponding angular attention output is given by

$$a_j := \frac{w_j}{Z}, \qquad Z := \sum_{i=1}^{N} w_i, \qquad \mathbf{y}^* := \sum_{j=1}^{N} a_j \boldsymbol{b}_j.$$

Our analysis targets an approximation of $\mathbf{y}^*$. Prior work shows that cosine-similarity kernels amenable to LSH closely approximate softmax attention in practice (Joshi et al., 2026), so we analyze this surrogate. Sampling is used only for analysis, enabling unbiased estimators and high-probability error bounds; the inference procedures in Algorithms 1 and 2 remain unchanged, with only the final aggregation replaced by a sampling-based estimator. In practice (Section 6), Algorithm 3 performs deterministic top-k selection using the same soft LSH scores, followed by standard softmax over the selected subset. Thus, the implementation preserves the pre-trained attention mechanism after retrieval, while the theory analyzes normalized soft LSH scores as a proxy attention distribution to study the SOCKET scoring kernel. We leave empirical evaluation of the sampling approach for future work.

5.1 Sampling-based Estimator

Our sampling-based estimator is constructed from the same scores produced in Algorithm 3, namely the soft-LSH weights $\widehat{w}_1, \ldots, \widehat{w}_N$. For the purpose of theoretical analysis, we rescale these scores *by the number of hash tables* L to obtain $\widetilde{w}_j := \frac{1}{L}\sum_{\ell=1}^{L} p_\tau^{(\ell)}\big(b_j^{(\ell)} \mid \mathbf{q}\big) = \frac{1}{L}\,\widehat{w}_j$ for $j = 1, \ldots, N$. To interpret the rescaled scores as an attention distribution, we further normalize $\widetilde{a}_j := \widetilde{w}_j/\widetilde{Z}$ with $\widetilde{Z} := \sum_{j=1}^{N} \widetilde{w}_j$ for $j = 1, \ldots, N$. The normalized weights $\widetilde{a}_j$ form a probability distribution over keys and serve as our proxy for angular attention coefficients in the theoretical analysis and the resulting

attention output $\mathbf{y}_{\tau,L}(\mathbf{q}) = \sum_{j=1}^{N} \widetilde{a}_j \boldsymbol{b}_j$. Note that the randomness in $\mathbf{y}_{\tau,L}(\mathbf{q})$ is due to the random hyperplanes for each hash table that we introduced. Consequently, we define the population (single-table) soft-count weight $w_{\tau,j} := \mathbb{E}\big(s_j^{(1)}(\mathbf{q})\big)$, the denominator $Z_\tau := \sum_{j=1}^{N} w_{\tau,j}$, and the output $\mathbf{y}_\tau(\mathbf{q}) := \sum_{j=1}^{N} a_{\tau,j} \boldsymbol{b}_j$, where $a_{\tau,j} = w_{\tau,j}/Z_\tau$. Now, we define a sampling distribution $p_j \propto \widetilde{a}_j \|\boldsymbol{b}_j\|_2$, and consider an estimator formed by drawing M independent samples $J_1, \ldots, J_M \sim p := (p_1, \ldots, p_N)$ and aggregating

$$\mathbf{T}(\mathbf{q}) := \frac{1}{M} \sum_{m=1}^{M} \frac{\widetilde{a}_{J_m}}{p_{J_m}} \boldsymbol{b}_{J_m}. \tag{5}$$

In the following section, we analyze $\mathbf{T}(\mathbf{q})$ using concentration arguments from Randomized Numerical Linear Algebra (RandNLA) literature (Tropp, 2015; Woodruff, 2014) and bound its deviation from angular attention.

5.2 Theoretical Analysis

Assumption 1 *$Z \geq Z_{min} > 0$ and $Z_\tau \geq Z_{\tau,min} > 0$.*

Assumption 2 *For each table $\ell \in [L]$ and bucket $r \in [R]$, the number of elements are bounded* i.e.,

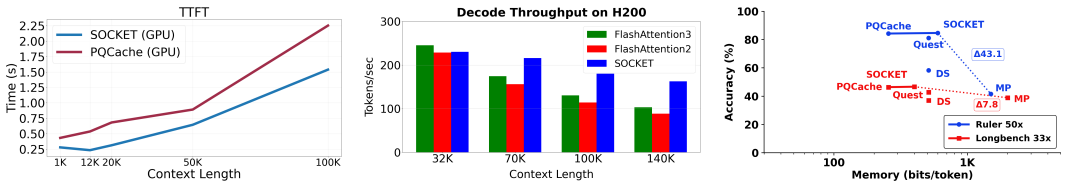
$$\#\left\{ j \in [N] : b_j^{(\ell)} = r \right\} \leq B.$$

Theorem 3 *Let $\mathbf{q} \in \mathbb{R}^d$ be a fixed query, with keys $\mathbf{k}_1, \ldots, \mathbf{k}_N \in \mathbb{R}^d$ and values $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N \in \mathbb{R}^d$. Under Assumptions equation 1 and equation 2, and for parameters L, M, and τ with $L \geq \frac{2B^2 \log(8/\delta)}{Z_{\tau,\min}^2}$, the estimator $\mathbf{T}(\mathbf{q})$ in eq. equation 5 satisfies*

$$\|\mathbf{T}(\mathbf{q}) - \mathbf{y}^*(\mathbf{q})\|_2 = \widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{L}} + \frac{1}{\sqrt{M}} + \varepsilon_\tau(\mathbf{q}) \right) \|\mathbf{V}\|_2,$$

with probability at least $1 - \delta$, where $\mathbf{y}^*(\mathbf{q})$ denotes the target (angular) attention output. $\widetilde{\mathcal{O}}(\cdot)$ hides a $\sqrt{\log(1/\delta)}$ factor and absolute constants depending only on $(B, Z_{\min}, Z_{\tau,\min})$, and is independent of N, d, L, and M. Moreover, $\varepsilon_\tau(\mathbf{q})$ quantifies the bias introduced by soft bucketization; it depends on τ and P (equivalently $R = 2^P$) and satisfies $\varepsilon_\tau(\mathbf{q}) \to 0$ as $\tau \to 0$ for fixed P.

Theorem 3 provides an end-to-end error decomposition for our proposed soft-count attention estimator $\mathbf{T}(\mathbf{q})$. The bound separates the total error into three terms corresponding to sampling variance, finite-table approximation error, and a bias induced by soft bucketization given by $\varepsilon_\tau = \mathbb{E}[1 - p_\tau^{(\ell)}(b_q \mid \mathbf{q})]$. This decomposition makes explicit how algorithmic parameters control different sources of error. Due to space constraints, we defer the proof of Theorem 3, parameter analysis, and further discussion to Appendix B.



(a) TTFT comparison (b) Throughput increase on H200 (c) Pareto Curve

Figure 3: (a) Time-to-first-token comparison on H100 for SOCKET and PQCache indexers. (b) Decode-only throughput as a function of context length for SOCKET (33× sparsity) and FlashAttention using GPT-FAST (Llama-3.1-8B). (c) Memory-Accuracy pareto curves with Ministral-3-8B.

5.3 Why Soft Collisions Yield More Stable Rankings than Hard LSH?

Our inference procedure relies on top-k selection from aggregated collision scores. Thus, the relevant criterion is ranking stability at finite L. When two keys have comparable scores, even small per-table fluctuations can change their ordering and lead to incorrect retrieval. Theorem 3 shows

Table 4: Comparison of sparse attention methods at 33× sparsity on LongBench across Llama-3.1-8B and Ministral-3-8B.[1]

Method	NQA	QAS	MFQA	HPQA	WIKI	MUS	GOV	QMSUM	MNews	LCC	Trivia	SamSUM	Retrieval	Repo	Avg
Llama-3.1-8B															
Quest (ps=16)	27.2	36.0	43.2	47.7	35.9	24.2	31.9	23.4	24.9	44.2	76.6	32.6	92.5	43.2	41.7
Quest (ps=8)	30.1	42.4	54.9	53.6	42.5	24.5	34.0	23.7	26.2	46.4	72.1	38.5	98.0	45.3	45.1
DS (l=16, ch=32)	26.8	38.5	47.5	48.7	43.2	19.2	32.6	23.4	25.3	42.2	68.9	30.7	97.0	39.1	41.6
HashAttn (b=128)	27.0	42.2	52.4	48.9	38.5	21.6	34.7	23.6	26.9	33.9	73.7	15.2	94.0	35.8	40.6
PQCache (pq=8, gr=4)	28.2	43.7	52.7	54.5	45.5	27.4	35.1	24.0	26.7	51.9	70.4	27.7	99.8	42.1	45.0
MagicPig (P=10, L=200)	21.8	26.1	31.1	41.2	23.6	15.5	28.3	23.1	23.3	41.9	62.4	28.3	48.9	37.7	32.4
SOCKET (P=8, L=50)	**29.3**	43.7	51.0	53.1	45.2	26.0	35.0	24.0	**27.0**	47.7	**79.1**	**34.5**	99.3	41.4	**45.5**
Ministral-3-8B															
Quest (ps=16)	23.1	38.6	33.2	38.4	36.0	28.2	22.6	17.9	20.1	37.8	85.0	31.1	97.9	28.0	38.4
Quest (ps=8)	27.3	43.1	42.7	56.0	48.2	27.3	23.7	18.5	20.4	42.8	88.6	33.6	99.7	28.6	42.9
DS (l=16, ch=32)	23.2	37.0	36.9	43.0	30.2	26.5	24.2	16.6	19.0	26.3	83.1	34.0	90.5	28.8	37.1
PQCache (pq=8, gr=4)	24.8	45.1	39.2	54.3	50.5	34.0	24.7	18.7	20.1	56.6	88.0	37.2	100.0	56.2	46.4
MagicPig (P=10, L=200)	22.5	33.8	32.7	45.7	27.8	24.7	23.0	17.3	19.0	46.7	82.0	34.6	88.1	46.8	38.9
SOCKET (P=8, L=50)	24.3	46.2	45.5	56.9	51.3	36.0	24.4	18.8	21.0	49.3	89.0	37.8	99.9	53.2	**46.7**

that soft bucketization introduces a controllable bias $\varepsilon_\tau(q)$, while the finite-table and sampling errors decay as $L^{-1/2}$ and $M^{-1/2}$. In the limit $\tau \to 0$, this bias vanishes and SOCKET recovers hard LSH. However, this limit also replaces smooth scores by discrete collision indicators, which are less informative for ranking at finite L.

Soft scores preserve directional structure: We first isolate the geometric effect of the per-plane scoring rule within a single hash table. Each random hyperplane produces a signed projection of the key, while the query determines a corresponding score. The next lemma shows that the resulting correlation with the true query–key similarity is governed by the alignment between the projected query and the score vector.

Lemma 4 *Fix a query $\mathbf{q} \in \mathbb{R}d$ with $\|\mathbf{q}\|_2 = 1$. Let $\{\hat{\mathbf{w}}_i\}_{i=1}^P \subset \mathbb{R}d$ be unit vectors with $\hat{\mathbf{w}}_i^\top \hat{\mathbf{w}}_j = 0$ for all $i \neq j$, and set $\mathbf{W} := [\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_P]^\top \in \mathbb{R}^{P\times d}$. Let $\mathbf{k} \sim \mathcal{N}(0, I_d)$ be an independent Gaussian key. For any deterministic scores $s_i = s_i(\mathbf{q}, \hat{\mathbf{w}}_i) \in \mathbb{R}$, define $X := \mathbf{q}^\top \mathbf{k}$ and $Y := \sum_{i=1}^P \operatorname{sign}(\hat{\mathbf{w}}_i^\top \mathbf{k}) s_i$. Then $\mathbb{E}[X] = \mathbb{E}[Y] = 0$, $\mathbb{E}[X^2] = 1$, $\mathbb{E}[Y^2] = \sum_{i=1}^P s_i^2$, and $\mathbb{E}[XY] = C \sum_{i=1}^P (\mathbf{q}^\top \hat{\mathbf{w}}_i) s_i$ with $C = \mathbb{E}|r| = \sqrt{2/\pi}$, $r \sim \mathcal{N}(0,1)$. Consequently, the correlation between the true similarity signal X and per-table–aggregated hash score Y is given by $\Gamma := C\, \mathbf{q}^\top \mathbf{W}^\top \hat{\mathbf{s}}$, where $\mathbf{s} := (s_1, \ldots, s_P)^\top$, $\hat{\mathbf{s}} := \mathbf{s}/\|\mathbf{s}\|_2$.*

Lemma 4 gives a general correlation formula for any per-plane score vector $\mathbf{s}$. It shows that the scoring rule affects ranking through the alignment of the normalized score vector $\hat{\mathbf{s}}$ with the projected query $\mathbf{Wq}$. For hard LSH, the score vector is sign-based and therefore discards magnitude information in the projected coordinates. In contrast, SOCKET's soft scores vary smoothly with these coordinates and, in the small-signal regime typical of high-dimensional random projections, are approximately linear in $\mathbf{Wq}$. Thus, soft scores preserve more directional information, which helps explain their more stable finite-L rankings. See Table 3 and Appendix C for more details.

6 Experiments

To ensure broad coverage across models supported by all baselines, we evaluate SOCKET on long-context models spanning different scales: Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Ministral-3-8B-Instruct-2512 (Liu et al., 2026), Qwen3-8B (Yang et al., 2025), Qwen3-4B-Instruct-2507 (Yang et al., 2025), and Qwen3-30B-A3B (Yang et al., 2025), all supporting contexts of up to 128K tokens. We evaluate on two complementary long-context benchmarks: **LongBench** (Bai et al., 2024), which covers question answering, reasoning, summarization, and code understanding, and **RULER** (Hsieh et al., 2024), which tests retrieval of sparse, position-sensitive information from long contexts. We implement SOCKET in the open-source Skylight framework[2] and run experiments on an NVIDIA H200 GPU. Following vAttention (Desai et al., 2026), we use dense attention during context processing and sparse attention during question processing and decoding. Under this protocol, MagicPIG (Chen et al., 2025) results differ from those originally reported; we discuss this

[1] At 33×: Ministral - HPQA/LCC/REPO use $(P, L) = (8, 55)$; Llama - LCC/REPO use $(P, L) = (10, 70)$.
[2] https://github.com/skylight-org/sparse-attention-hub

effect in Appendix A.3. For a fair comparison, all baselines are fully sparse across layers, with no dense layers retained. Following common sparse-attention practice, we additionally retain a small set of sink and local-window tokens (e.g., 64 or 128 tokens).

Baselines: We compare SOCKET against five representative sparse attention methods: MagicPig (Chen et al., 2025), HashAttention (Desai et al., 2025), Quest (Tang et al., 2024), Double Sparsity (Yang et al., 2024), and PQCache (Zhang et al., 2025). These methods cover a broad spectrum of recent top-k-based and sampling-based sparsification approaches. For all baselines, we use the hyperparameter settings recommended by the respective authors to ensure a fair comparison.

6.1 Effectiveness of SOCKET Across Models

Across model families and sparsity regimes, SOCKET consistently matches or outperforms existing baselines. On **Llama-3.1-8B**, SOCKET outperforms Quest and PQCache on LongBench across all evaluated sparsity levels (Tables 4 and 10), achieving 49 at $10\times$ sparsity, approximately 2-3 points above the strongest baseline, and remaining competitive at $33\times$ sparsity with a score of 45.5. On RULER-HARD-32K, SOCKET similarly outperforms the baselines for both Llama-3.1-8B and Ministral-3-8B at $5\times$, $10\times$, and $20\times$ sparsity, while remaining competitive with strong data-dependent methods even at $50\times$ sparsity (Tables 1 and 8). These results highlight the effectiveness of SOCKET's soft LSH-based scoring under aggressive sparsification. This trend generalizes to **Qwen3-8B** and **Ministral-3-8B**, where SOCKET matches or outperforms PQCache and Quest across sparsity levels (Tables 4, 9, and 10). Overall, SOCKET establishes a stronger accuracy–memory Pareto frontier for data-agnostic methods, particularly under light memory budgets (fig. 3c). Additional results across model scales in the Appendix (Tables 11 and 12) further show that SOCKET can retain performance close to dense attention even at extreme sparsity.

6.2 Efficiency of SOCKET

SOCKET's decode step avoids reading the full key cache to determine which positions to attend to. During prefill, each key is hashed once into L bucket codes of P hyperplanes, which are stored alongside the KV cache and its value norm. At decode, a fused CUDA kernel projects each query onto the hash planes and computes a soft bucket distribution over the 2^P buckets for each table. A second kernel streams the stored codes and scores each cached position using its value-norm-weighted soft-collision evidence, $\sum_\ell p_\ell[b_\ell(x_j)]$. Under grouped-query attention, codes are stored once per KV head and code loads are reused across query heads, reducing memory traffic in the bandwidth-bound scoring kernel. An exact radix-threshold selection then identifies the top-k scores in a fixed number of streaming passes without sorting. Finally, attention over the selected keys is computed using a two-stage Flash Decode Triton kernel. We evaluate SOCKET's decoding efficiency on an NVIDIA H200 GPU using decode-only throughput, extending GPT-FAST with custom Triton/CUDA kernels for sparse attention. As shown in fig. 3b, SOCKET's gains increase with context length: it surpasses FlashAttention3 (Shah et al., 2024) at 70K tokens, with speedup increasing from approximately $1.18\times$ at 70K to $1.3\times$ at 100K tokens. At 140K tokens, SOCKET achieves approximately $1.5\times$ speedup at $33\times$ sparsity, demonstrating increasingly favorable efficiency for long-context decoding.

7 Conclusion

We introduce SOCKET, a SOft Collision Kernel EsTimator for stable and efficient key ranking in sparse attention. Across LongBench and RULER, SOCKET achieves strong performance across three instruction-tuned model families while offering a favorable memory–accuracy tradeoff among data-agnostic methods. Our optimized CUDA/Triton kernels achieve up to $1.5\times$ higher decoding throughput than FlashAttention in GPT-FAST, with only $\sim$15% additional memory beyond the standard KV cache for hash-table storage. Notably, SOCKET remains effective with representations as compact as 100 bits/token on aggregation tasks, suggesting that highly compressed representations may suffice for reliable KV selection. Further exploring this low-memory regime and extending SOCKET to hybrid architectures are promising directions for future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with polishing the writing and exploring design considerations for SOCKET's kernel implementations. Generative AI tools were not used to develop the core idea of SOCKET, formulate its methodology, conduct experiments, analyze results, or draw scientific conclusions. We carefully reviewed all AI-assisted content and take full responsibility for the final content of this work, including all text, claims, results, and artifacts produced.

ETHICS STATEMENT

Our work focuses on accelerating model inference through sparse attention. As such, it has the potential for broader impact by enabling practitioners to perform long-context inference more efficiently and at lower computational cost. Our experiments rely exclusively on publicly available datasets to evaluate the performance of our method, and we do not identify any additional ethical considerations arising specifically from this work.

REPRODUCIBILITY STATEMENT

We provide the source code and configurations for our key experiments on RULER and LongBench, along with instructions for reproducing our results, in Section 6. All theoretical results and proofs are presented in the appendix, together with detailed explanations and the underlying assumptions. We have thoroughly validated our implementation and empirically verified that the proposed algorithm behaves as expected.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, August 2024.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The Long-Document Transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, 2020.

Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC'02)*, pp. 380–388. ACM, 2002.

Beidi Chen and Anshumali Shrivastava. Densified Winner Take All (WTA) Hashing for Sparse Datasets. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

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.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sashank Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling Language Modeling with Pathways. *Journal of Machine Learning Research*, 24(1), 2023.

Benjamin Coleman and Anshumali Shrivastava. Sub-linear RACE Sketches for Approximate Kernel Density Estimation on Streaming Data. In *WWW '20: The Web Conference*, 2020.

Benjamin Coleman, Richard Baraniuk, and Anshumali Shrivastava. Sub-linear Memory Sketches for Near Neighbor Search on Streaming Data. In *International Conference on Machine Learning*, 2020.

Aditya Desai, Shuo Yang, Alejandro Cuadron, Ana Klimovic, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. HashAttention: Semantic Sparsity for Faster Inference. In *International Conference on Machine Learning*, 2025.

Aditya Desai, Kumar Krishna Agrawal, Shuo Yang, Alejandro Cuadron, Luis Gaspar Schroeder, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. vAttention: Verified Sparse Attention via Sampling. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=zzTDulLys0.

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)*, 2019.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A Survey on In-Context Learning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model Tells You What to Discard: Adaptive KV Cache Compression for LLMs. In *International Conference on Learning Representations*, 2024.

Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *Very Large Data Bases (VLDB)*, volume 99, pp. 518–529, 1999.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024.

Ankit Gupta, Guy Dar, Shaya Goodman, David Ciprut, and Jonathan Berant. Memory-efficient Transformers via Top-k Attention. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pp. 39–52, 2021.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. Training Compute-Optimal Large Language Models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. https://openreview.net/forum?id=iBBcRUlOAPR.

Coleman Richard Charles Hooper, Sehoon Kim, Hiva Mohammadzadeh, Monishwaran Maheswaran, Sebastian Zhao, June Paik, Michael W Mahoney, Kurt Keutzer, and Amir Gholami. Squeezed Attention: Accelerating Long Context Length LLM Inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024.

Piotr Indyk and Rajeev Motwani. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604–613, 1998.

Sahil Joshi, Agniva Chowdhury, Amar Kanakamedala, Ekam Singh, Evan Tu, and Anshumali Shrivastava. RACE Attention: A Strictly Linear-Time Attention Layer for Training on Outrageously Large Contexts. In *The Fourteenth International Conference on Learning Representations*, 2026.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature*, 596:583–589, 2021.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The Efficient Transformer. In *International Conference on Learning Representations*, 2020.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. *International Conference on Machine Learning*, 2023a.

Raymond Li, Loubna Ben allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia LI, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Joel Lamy-Poirier, Joao Monteiro, Nicolas Gontier, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Ben Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason T Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Urvashi Bhattacharyya, Wenhao Yu, Sasha Luccioni, Paulo Villegas, Fedor Zhdanov, Tony Lee, Nadav Timor, Jennifer Ding, Claire S Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro Von Werra, and Harm de Vries. Starcoder: may the source be with you! *Transactions on Machine Learning Research*, 2023b. ISSN 2835-8856.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, et al. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=8z3cOVER4z.

Hao Liu, Matei Zaharia, and Pieter Abbeel. RingAttention with Blockwise Transformers for Near-Infinite Context. In *The Twelfth International Conference on Learning Representations*, 2024.

Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Hashing with Graphs. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

Iosif Pinelis. An Approach to Inequalities for the Distributions of Infinite-Dimensional Martingales. In *Probability in Banach Spaces, 8: Proceedings of the Eighth International Conference*, pp. 128–134. Springer, 1992.

Ofir Press, Noah Smith, and Mike Lewis. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. In *International Conference on Learning Representations*, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models from Natural Language Supervision. In *Proceedings of 38th International conference on machine learning*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. Parallel Context Windows for Large Language Models. In *Proceedings of the 61st annual meeting of the association for computational linguistics*, 2023.

Ludan Ruan and Qin Jin. Survey: Transformer based Video-Language Pre-training. *AI Open*, 3:1–13, 2022. https://www.sciencedirect.com/science/article/pii/S2666651022000018.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and Accurate Attention with Asynchrony and Low-precision. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Prajwal Singhania, Siddharth Singh, Shwai He, Soheil Feizi, and Abhinav Bhatele. Loki: Low-rank Keys for Efficient Sparse Attention. In *Advances in Neural Information Processing Systems*, 2024.

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

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. QUEST: Query-Aware Sparsity for Efficient Long-Context LLM Inference. In *International Conference on Machine Learning*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*, 2023.

Joel A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1–2):1–230, 2015. ISSN 1935-8237. doi: 10.1561/2200000048. URL https://doi.org/10.1561/2200000048.

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*, 2017.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *In Advances in Neural Information Processing Systems*, 2022.

David P. Woodruff. Sketching as a Tool for Numerical Linear Algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014. ISSN 1551-305X. doi: 10.1561/0400000060. URL https://doi.org/10.1561/0400000060.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. InfLLM: Training-Free Long-Context Extrapolation for LLMs with an Efficient Context Memory. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient Streaming Language Models with Attention Sinks. In *International Conference on Learning Representations*, 2024b.

Jay Yagnik, Dennis Strelow, David A Ross, and Ruei-sung Lin. The Power of Comparative Reasoning. In *2011 International Conference on Computer Vision*, 2011.

Siyuan Yan, Guo-Qing Jiang, Yuchen Zhang, Xiaoxing Ma, Ran Zhu, Chun Cao, and Jingwei Xu. Adamas: Hadamard Sparse Attention for Efficient Long-Context Inference. *arXiv preprint arXiv:2510.18413*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025.

Shuo Yang, Ying Sheng, Joseph E. Gonzalez, Ion Stoica, and Lianmin Zheng. Post-Training Sparse Attention with Double Sparsity. *arXiv preprint arXiv:2408.07092*, 2024.

Hailin Zhang, Xiaodong Ji, Yilin Chen, Fangcheng Fu, Xupeng Miao, Xiaonan Nie, Weipeng Chen, and Bin Cui. PQCache: Product Quantization-based KVCache for Long Context LLM Inference. *Proceedings of the ACM on Management of Data*, 3(3):1–30, 2025.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark W. Barrett, Zhangyang Wang, and Beidi Chen. H_2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. In *Advances in Neural Information Processing Systems*, 2023.

Xuanhe Zhou, Zhaoyan Sun, and Guoliang Li. DB-GPT: Large Language Model Meets Database. *Data Science and Engineering*, 9:102 – 111, 2024.

Kan Zhu, Tian Tang, Qinyu Xu, Zhan Jin, Yile Gu, Zhichen Zeng, Rohan Kadekodi, Liangyu Zhao, Ang Li, Arvind Krishnamurthy, and Baris Kasikci. Tactic: Adaptive Sparse Attention with Clustering and Distribution Fitting for Long-Context LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026.

A Appendix

A.1 Ablations for SOCKET

We ablate the key SOCKET hyperparameters on five RULER-32K-Hard datasets (`nm2`, `qa1`, `vt`, `nm3`, `qa2`) under the 20× sparsity setting for Llama-3.1-8B-Instruct. We study the number of hyperplanes P, the number of hash tables L, and the soft-hashing temperature τ. The final column reports the average score.

P	nm2	qa1	vt	nm3	qa2	Avg
4	68	53	79.0	1	40	48.20
5	83	65	78.8	15	50	58.36
6	91	73	85.4	52	53	70.88
7	93	73	87.6	84	52	77.92
8	95	75	92.0	91	51	80.80
9	96	77	93.4	91	52	81.88
10	96	78	96.4	94	52	83.28

(a) Varying P; $\tau = 0.4$, $L = 60$.

L	nm2	qa1	vt	nm3	qa2	Avg
10	75	61	79.2	31	48	58.84
20	87	71	86.6	67	50	72.32
40	93	78	91.2	92	50	80.84
60	98	76	92.8	93	51	82.16
70	96	76	93.0	93	53	82.20

(b) Varying L; $\tau = 0.5$, $P = 10$.

τ	nm2	qa1	vt	nm3	qa2	Avg
0.1	85	71	78.8	71	49	70.96
0.2	94	78	93.6	95	50	82.12
0.3	94	82	90.0	96	52	82.80
0.4	96	78	96.4	94	52	83.28
0.5	98	76	92.8	93	51	82.16
0.6	95	75	90.6	84	51	79.12
0.7	92	76	85.2	67	53	74.64
0.8	89	68	85.2	29	51	64.40

(c) Varying τ; $P = 10$, $L = 60$.

Table 5: Hyperparameter ablations for SOCKET on RULER-32K-Hard at 20× sparsity using Llama-3.1-8B-Instruct.

We observe consistent trends across all three hyperparameters. Increasing P improves discrimination between buckets and steadily increases accuracy, with gains saturating beyond $P = 9$. Similarly, increasing the number of hash tables L improves recall by reducing approximation error, though the marginal gains diminish after $L = 60$. For the temperature parameter τ, intermediate values ($\tau \in [0.3, 0.5]$) achieve the best performance, balancing the bias–variance trade-off between hard and overly smooth soft hashing. These empirical findings are consistent with Theorem 3. In particular, the finite-L approximation error decreases as L increases, while moderate τ preserves score discrimination without introducing instability. Based on this trade-off, we use $P = 10$, $L = 60$, and $\tau \in [0.3, 0.5]$ in all main experiments.

A.2 Ablations for Hard LSH Estimator

In this section, we evaluate Hard LSH under the same memory budget as SOCKET (600 bits per token) on five RULER32K-Hard datasets (`nm2`, `qa1`, `vt`, `nm3`, and `qa2`) at 20× sparsity using Llama-3.1-8B-Instruct. The corresponding SOCKET results under the same budget are provided in Section A.1.

P	nm2	qa1	vt	nm3	qa2	Avg
1	40	46	71.2	3	44	40.84
2	70	63	79.4	30	47	57.88
3	74	63	80.6	22	49	57.72
4	62	60	72.4	20	49	52.68
5	43	58	63.2	5	48	43.44

(a) Varying P ($L = 60$).

L	Bits	nm2	qa1	vt	nm3	qa2	Avg
70	140	76	69	81.4	33	47	61.28
100	200	85	71	85.6	66	49	71.32
150	300	93	73	89	80	51	77.20
200	400	94	76	91	88	50	79.80
250	500	94	76	94	93	51	81.60
300	600	97	81	95	94	53	84

(b) Varying L (fixed $P = 2$).

L	Bits	nm2	qa1	vt	nm3	qa2	Avg
350	700	97	81	95.2	94	53	84.04
400	800	97	81	95.8	93	52	83.76
450	900	96	79	95.8	91	50	82.36
500	1000	98	82	96.0	95	53	84.8

(c) Larger memory budget.

Table 6: Hard LSH ablations on RULER32K-Hard at 20× sparsity (Llama-3.1-8B-Instruct).

Hard LSH shows strong sensitivity to the hyperparameter P. The best performance is achieved at $P = 2$, while larger values substantially degrade accuracy due to increasingly sparse and brittle bucket assignments. Unlike SOCKET, which leverages soft bucket probabilities, hard LSH relies on discrete collisions and therefore requires substantially larger L to recover comparable retrieval quality. Under the same memory budget of 600 bits/token, hard LSH reaches an average score of 84, still below SOCKET's 85.08. Even when the memory budget is increased to 1000 bits/token, Hard LSH reaches 84.80, which remains lower than SOCKET under significantly smaller memory usage.

A.3 COMPARISON WITH MAGICPIG

The variation in the reported results relative to the original MagicPIG (Chen et al., 2025) primarily arises from differences in the evaluation protocol. As discussed in vAttention (Desai et al., 2026) (Table 10), two evaluation setups are commonly used for sparse attention methods.

Setup A: In this setup, the full context and question are first processed using dense attention, and sparsity is applied only during the decoding stage. This is the setup used in MagicPIG (Chen et al., 2025).

Setup B: In this setup, sparsity is applied during both question processing and decoding. This is the setup used in our evaluation. Compared to Setup A, this setting is substantially more challenging, since the sparse method must not only support efficient decoding but also retrieve relevant information while processing the question itself.

In addition, MagicPig retains (0, 16) dense layers as a fallback mechanism. To ensure a fair comparison under a fully sparse setting, we remove all dense layers and evaluate MagicPig as a completely sparse method in Tables 1 and 8, applying sparsity uniformly across every layer. We additionally evaluate a hybrid dense-sparse variant that retains the 0th and 16th layers as dense, consistent with the original MagicPig design, while making the remaining layers correspondingly sparser to preserve a comparable overall sparsity ratio. The results are shown in Table 7.

Table 7: Comparison with MagicPIG under different evaluation settings using Llama-3.1-8B.

Method	Sparsity	nm2	nm3	vt	qa1	qa2	Avg
MagicPIG (0,16 dense)	5×	77	71	96.8	69.6	46	72.1
MagicPIG (0,16 dense)	10×	35	16	95.8	53	46	49.2
MagicPIG (0,16 dense)	50×	0.6	0	81.8	24	32	27.7
MagicPIG (fully sparse)	5×	49	23	93.3	63	47	55.1
MagicPIG (fully sparse)	10×	10.3	0	93	47	42	38.5
MagicPIG (fully sparse)	50×	0.3	0	70.6	21	28	24.0
SOCKET	5×	97	100	95.2	81	53	**85.2**
SOCKET	10×	96	99.6	94.2	80	52	**84.4**
SOCKET	50×	80	73	83.8	77	50	**72.8**

As expected, the hybrid dense-sparse MagicPig configuration performs better than its fully sparse counterpart. However, even under this more favorable setting, MagicPIG remains consistently below SOCKET across all sparsity levels.

A.4 Additional Experiments and Error Bars

Table 8: Performance on RULER-HARD-32K using Ministral-3-8B and Llama-3.1-8B. Runs for fwe use $(P = 10, L = 10, \tau = 1e^{-3})$.

Method	Spr	nm2	nm3	vt	fwe	qa1	qa2	Avg
Ministral-3-8B-Instruct-2512								
Dense	–	99	100	100	94.3	79	63	89.2
Quest (ps=32)	5×	99	89	100	96.3	79	60	87.2
DS (l=8, ch=32)	5×	99	99	100	93	76	62	88.2
PQCache (pq=8, gr=4)	5×	99	100	100	94.3	79	57	88.2
MagicPig (P=7, L=100)	5×	85.3	60.3	100	94.3	77.3	59.6	79.5
SOCKET (P=10, L=60)	5×	**99**	**100**	**100**	**97.6**	**80**	**63**	**89.9**
Quest (ps=32)	10×	85	39	100	97	81	60	77.0
DS (l=8, ch=32)	10×	97	81	100	92.3	76	56	83.7
PQCache (pq=8, gr=4)	10×	99	99	100	92.6	79	63	88.8
MagicPig (P=8, L=120)	10×	47.6	11	99.8	93.4	69.6	54	62.6
SOCKET (P=10, L=60)	10×	**99**	98	**100**	**98**	**81**	61	**89.5**
Llama-3.1-8B-Instruct								
Dense	–	96	100	95.2	93.3	82	52	86.4
Quest (ps=32)	5×	91	92	97.2	90.3	82	52	84.1
DS (l=8, ch=32)	5×	93	100	97.0	92.0	81	52	85.8
HashAttn (b=128)	5×	92	99	90.4	93.3	82	50	84.5
PQCache (pq=8, gr=4)	5×	97	99	95.2	95.4	81	51	86.4
MagicPig (P=7, L=100)	5×	49	23	93.3	92.4	63	47	61.3
SOCKET (P=10, L=60)	5×	**97**	**100**	95.2	**96**	81	**53**	**87.0**
Quest (ps=32)	10×	86	45	96	90	80	53	75.0
DS (l=8, ch=32)	10×	76	95	94	90.6	72	50	79.6
HashAttn (b=128)	10×	87	89	85.6	92.6	79	52	80.9
PQCache (pq=8, gr=4)	10×	98	98	94.0	92.3	81	51	85.7
MagicPig (P=8, L=120)	10×	10.3	0	93.0	92.0	47	42	47.4
SOCKET (P=10, L=60)	10×	96	**99.6**	**94.2**	**96.4**	80	**52**	**86.4**

Table 9: Comparison of dense and sparse attention methods on LongBench (Qwen3-8B).[1]

Method	Sparsity	NQA	QAS	MFQA	HPQA	WIKI	MUS	GOV	QMSUM	MNews	LCC	Trivia	SamSUM	Retrieval	Repo	Avg
Baseline	Dense	26.5	43.8	55.7	62.3	48.0	35.3	33.5	24.5	24.6	65.8	90.3	40.0	91.6	62.2	**50.3**
Quest (ps=32)	10×	24.4	37.3	51.3	57.6	44.7	33.1	32.7	23.6	23.2	65.8	85.8	38.8	95.0	62.7	48.3
DS (l=8, ch=32)	10×	22.1	35.0	46.6	53.3	43.5	25.9	31.2	22.2	23.2	67.7	85.7	38.3	95.6	60.0	46.5
PQCache (pq=8, gr=4)	10×	26.0	43.0	53.8	63.3	48.7	33.5	33.8	24.2	24.6	68.1	89.6	42.3	94.5	62.3	50.6
MagicPig (P=8, L=120)	10×	25.5	41.4	52.1	58.4	43.9	31.1	32.7	23.9	24.2	65.5	84.4	39.6	94.4	58.8	48.3
SOCKET (P=8, L=50)	10×	24.0	42.2	**55.7**	**63.5**	48.5	**35.0**	**34.5**	24.0	**24.7**	67.3	88.8	41.8	**98.7**	61.9	**50.8**
Quest (ps=16)	33×	22.1	26.3	43.5	51.1	35.2	27.4	30.4	21.4	22.3	50.3	77.8	35.8	85.2	50.1	41.4
Quest (ps=8)	33×	26.6	37.9	51.8	58.0	45.9	28.4	32.0	23.1	22.9	55.2	80.5	37.4	93.3	57.0	46.4
DS (l=16, ch=32)	33×	11.3	16.9	31.2	30.1	25.7	8.3	24.2	18.5	21.0	44.6	57.5	26.3	27.1	40.8	27.4
PQCache (pq=8, gr=4)	33×	24.7	39.4	52.6	61.6	47.4	30.0	34.1	24.0	24.5	66.1	84.0	42.1	92.4	59.1	48.7
MagicPig (P=10, L=200)	33×	21.5	36.7	49.6	52.4	38.9	28.1	32.4	23.6	23.7	61.1	75.3	36.7	90.4	53.4	44.6
SOCKET (P=8, L=50)	33×	23.6	39.4	52.3	61.0	47.2‡	**32.6**	**34.8**	23.6	**26.0**	61.9‡	**86.3**	40.8‡	**98.8**	54.8‡	**49.2**

Table 10: Comparison of sparse attention methods at 10× sparsity on LongBench across Llama-3.1-8B and Ministral-3-8B.

Method	NQA	QAS	MFQA	HPQA	WIKI	MUS	GOV	QMSUM	MNews	LCC	Trivia	SamSUM	Retrieval	Repo	Avg
Llama-3.1-8B															
Quest (ps=32)	30.0	42.1	53.9	51.2	43.1	27.1	33.9	24.2	25.2	53.1	74.6	40.1	99.5	49.3	46.2
DS (l=8, ch=32)	28.9	47.1	53.9	53.6	46.6	25.4	35.1	24.6	26.4	53.6	79.9	21.9	99.5	43.8	45.7
HashAttn (b=128)	28.8	45.7	55.6	53.0	45.9	25.4	35.1	24.7	26.7	51.6	76.1	19.9	100.0	44.0	45.2
PQCache (pq=8, gr=4)	28.8	45.9	54.0	58.7	52.0	27.2	35.0	25.2	26.8	55.5	76.9	26.4	99.5	44.3	46.8
MagicPig (P=8, L=120)	26.8	38.2	45.4	49.1	35.1	23.3	30.7	24.3	25.1	48.6	73.1	32.8	85.6	43.0	41.5
SOCKET (P=8, L=50)	**30.5**	**48.5**	**56.2**	56.9	49.3	30.0	**35.6**	25.0	**27.0**	54.5	**82.6**	**42.6**	**100.0**	**47.4**	**49.0**
Ministral-3-8B															
Quest (ps=32)	27.3	43.8	42.1	58.4	50.7	35.9	23.5	18.6	19.4	41.1	88.1	34.5	100.0	46.0	45.0
DS (l=8, ch=32)	25.1	43.7	42.9	56.2	51.4	39.0	23.6	18.0	19.6	46.5	88.3	36.0	100.0	43.0	45.2
PQCache (pq=8, gr=4)	27.4	44.2	44.1	60.1	55.8	37.5	24.4	18.4	20.1	60.0	89.8	37.8	99.9	61.4	48.6
MagicPig (P=8, L=120)	25.9	41.6	40.8	54.2	47.4	36.0	23.8	17.2	19.8	52.5	88.3	36.4	99.4	54.9	45.6
SOCKET (P=8, L=50)	26.6	46.2	45.0	59.5	56.2	37.1	24.5	18.6	20.5	58.6	89.6	37.9	99.9	61.7	**48.7**

Spr	vt	qa1	qa2	fwe	nm2	nm3	Avg
Dense	99.8	88	59	100	99.3	100	91.0
5×	100	88	60	98.6	100	100	**91.1**
10×	100	87	62	97.6	100	100	**91.1**
20×	100	87	60	97.3	100	100	90.7
50×	100	85	57	95.6	100	100	89.6

Table 11: RULER-HARD-32K performance using Qwen3-30B-A3B. Sparse results use SOCKET ($P = 10$, $L = 60$, $\tau = 0.3$).

Spr	vt	qa1	qa2	fwe	nm2	nm3	Avg
Dense	100	79	60	95	99	99	**88.7**
5×	100	78	60	94	99	99	88.3
10×	100	78	53	92.3	99	98	86.7
20×	100	76	52	92	99	99	86.3
50×	100	68	52	91	92	76	79.8

Table 12: RULER-HARD-32K performance using Qwen3-4B-Instruct-2507. Sparse results use SOCKET ($P = 10$, $L = 60$, $\tau = 0.3$).

Spr	vt	qa1	qa2	fwe	nm2	nm3
5×	0	0.5	0	0	0.5	0
10×	0	2.0	0.5	0.5	0	0
20×	0	1.5	0	0.2	1.0	0
50×	0	1.1	1.2	1.0	0.5	2.0

Table 13: Standard deviations for SOCKET ($P = 10$, $L = 60$) using Qwen3-30B-A3B.

Spr	vt	qa1	qa2	fwe	nm2	nm3
5×	0	0.5	0	0	0.5	0
10×	0	2.0	0.5	0.5	0	0
20×	0	1.5	0	0.2	1.0	0
50×	0	1.1	1.2	1.0	0.5	2.0

Table 14: Standard deviations for SOCKET ($P = 10$, $L = 60$) using Qwen3-4B-Instruct-2507.

Method	nm2	nm3	vt	fwe	qa1	qa2
5× sparsity						
PQCache	0	0	0	0.1	0	0
MagicPig	3.0	2.5	0.5	1.6	1.0	1.7
SOCKET	0.3	0	0.3	0	1.5	0.5
10× sparsity						
PQCache	0.5	0.5	0.1	0.1	0.5	0.5
MagicPig	3.2	0	0.6	1.5	2.2	1.7
SOCKET	0.5	0.5	0.3	0.5	2.6	0.5
20× sparsity						
PQCache	2.0	1.0	0.2	0.3	0	1.0
MagicPig	1.0	0	2.9	2.3	1.5	2.8
SOCKET	1.7	1.5	1.3	1.3	2.5	0.5
50× sparsity						
PQCache	2.0	0.5	0.7	1.1	2.0	1.7
MagicPig	0.5	0	2.5	0.8	1.1	2.0
SOCKET	3.0	3.4	1.7	1.5	1.1	1.0

Table 15: Standard deviations across 3 different seeds for Tables 1 and 8 (Llama-3.1-8B).

Method	nm2	nm3	vt	fwe	qa1	qa2
5× sparsity						
PQCache	0	0	0	0.2	0.5	0
MagicPig	3.0	0.5	0	1.3	1.5	1.5
SOCKET	0.5	0	0	0.4	1.0	1.5
10× sparsity						
PQCache	0	0	0	0.2	1.2	0.5
MagicPig	3.0	1.0	0	1.7	3.0	2.5
SOCKET	0	0	0	1.5	1.5	0.5
20× sparsity						
PQCache	0.5	0.5	0	0.2	1.2	0
MagicPig	1.7	0	0.4	3.0	2.5	3.1
SOCKET	0	2.3	0	0.5	1.5	0.5
50× sparsity						
PQCache	1.0	1.2	0	1.0	0.6	1.6
MagicPig	0	0	2.0	0.5	2.3	2.5
SOCKET	0	2.3	0	1.5	2	1.5

Table 16: Standard deviations across 3 different seeds for Tables 1 and 8 (Ministral-3-8B).

Table 17: Standard deviations across 3 different seeds for Llama-3.1-8B on LongBench.

Method	Sparsity	NQA	QAS	MFQA	HPQA	WIKI	MUS	GOV	QMSUM	MNews	LCC	Trivia	SamSUM	Retrieval	Repo
PQCache	10×	0.5	0.6	0.5	0.1	0.2	0.7	0.2	0.1	0.3	0.5	1.1	0.5	0.0	0.6
MagicPig	10×	1.1	0.8	0.6	0.4	2.4	2.4	0.6	0.1	0.2	1.3	1.8	2.1	3.0	2.0
SOCKET	10×	0.7	0.6	0.2	0.4	1.5	1.0	0.2	0.4	0.1	0.2	0.4	0.3	0.0	0.3
PQCache	33×	0.8	1.2	0.9	1.3	1.5	0.4	0.1	0.2	0.1	0.9	0.8	0.6	0.2	0.2
MagicPig	33×	0.8	1.4	1.4	3.0	1.4	1.6	0.2	0.2	0.1	0.9	3.0	2.1	3.0	1.9
SOCKET	33×	0.3	0.8	1.0	1.8	1.9	0.9	0.4	0.1	0.6	0.7	0.7	1.2	0.7	0.8

Table 18: Standard deviations across 3 different seeds for Qwen3-8B on Longbench.

Method	Sparsity	NQA	QAS	MFQA	HPQA	WIKI	MUS	GOV	QMSUM	MNews	LCC	Trivia	SamSUM	Retrieval	Repo
PQCache (pq=8, gr=4)	10×	0.6	0.3	0.1	0.3	0.6	0.5	0.1	0.0	0.0	0.0	0.3	0.1	0.5	0.2
MagicPig (P=8, L=120)	10×	0.1	1.4	0.6	0.9	2.0	1.6	0.0	0.3	0.0	0.1	1.9	0.5	1.4	1.3
SOCKET (P=8, L=50)	10×	0.5	0.5	0.4	0.7	0.2	0.3	0.0	0.1	0.2	0.7	0.2	0.2	0.5	0.8
PQCache (pq=8, gr=4)	33×	0.3	0.8	0.6	0.9	0.8	2.2	0.1	0.3	0.1	0.4	0.6	0.3	0.6	0.3
MagicPig (P=10, L=200)	33×	1.3	0.9	2.0	2.0	2.3	2.8	0.3	0.2	0.4	0.6	3.0	0.2	0.7	1.4
SOCKET (P=8, L=50)	33×	0.9	1.5	0.8	1.0	2.0	1.8	0.1	0.1	0.1	0.8	0.5	0.5	0.7	0.3

Table 19: Standard deviations across 3 different seeds for Ministral-3-8B on Longbench.

Method	Sparsity	NQA	QAS	MFQA	HPQA	WIKI	MUS	GOV	QMSUM	MNews	LCC	Trivia	SamSUM	Retrieval	Repo
PQCache (pq=8, gr=4)	10×	0.4	0.4	0.3	0.4	0.9	0.4	0.1	0	0.2	0.2	0.2	0.2	0.1	0.4
MagicPig (P=8, L=120)	10×	1.3	1.8	0.9	1.3	2.0	2.0	0.3	0.2	0.1	2.0	0.7	1.1	0.2	2.0
SOCKET (P=8, L=50)	10×	0.7	0.4	0.6	0.6	1.2	0.7	0.1	0.1	0.2	1.0	0.3	0.3	0.1	1.2
PQCache (pq=8, gr=4)	33×	0.2	0.3	0.8	1.1	1.2	0.4	0	0.2	0.3	0.2	0.6	0.7	0	0.3
MagicPig (P=10, L=200)	33×	1.4	0.4	1.0	2.0	2.0	0.3	0.4	0.2	0.3	1.5	0.3	0.7	1.7	0.7
SOCKET (P=8, L=50)	33×	0.6	1.2	0.5	1.3	1.6	1.3	0.2	0.2	0.1	1.2	0.7	0.6	0.1	2.4

Table 20: Hyperparameter settings used across datasets and models.

Dataset	Spr	P	L	τ	Sink/Recent	Models
LongBench	10×	8	50	0.3-0.7	64/64	Llama-3.1-8B, Ministral-3-8B, Qwen3-8B
LongBench	33×	8	50	0.3-0.7	64/16	Llama-3.1-8B, Ministral-3-8B, Qwen3-8B
RULER-HARD-32K	–	10	60	0.3-0.7	128/128	Llama-3.1-8B, Ministral-3-8B, Qwen3-4B, Qwen3-30B-A3B

A.5 Definitions of metrics used in Fig. 2

Normalized Discounted Cumulative Gain (NDCG): NDCG measures the quality of a ranked list by accounting for both item relevance and ranking position, assigning higher weight to relevant items appearing earlier. Given a ranked list of length k with relevance scores $\{r_i\}_{i=1}^{k}$, the discounted cumulative gain is defined as

$$\mathrm{DCG} = \sum_{i=1}^{k} \frac{2^{r_i} - 1}{\log_2(i+1)}.$$

Let IDCG denote the DCG obtained by sorting items in decreasing order of relevance. The normalized score is then

$$\mathrm{NDCG} = \frac{\mathrm{DCG}}{\mathrm{IDCG}}.$$

Precision: Precision measures the fraction of retrieved items that are relevant. Let S_k denote the set of top-k retrieved items and R the set of relevant items. Precision is defined as

$$\mathrm{Precision} = \frac{|S_k \cap R|}{k}.$$

Jaccard Similarity: Jaccard similarity measures the overlap between two sets. Given two sets A and B, it is defined as

$$\mathrm{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

B Proof of Theorem 3

B.1 Additional notes on Theorem 3

Theorem 3 provides an end-to-end error decomposition for the proposed soft-count attention estimator $\mathbf{T}(\mathbf{q})$. The bound separates the total error into three components: (i) sampling variance arising from the value-aware sampling step, (ii) finite-table approximation error due to using a limited number of hash tables, and (iii) a bias induced by soft bucketization, quantified by $\varepsilon_\tau = \mathbb{E}\Big[1 - p_\tau^{(\ell)}(b_q \mid \mathbf{q})\Big]$. This decomposition makes explicit how the algorithmic parameters jointly control the different sources of error. To provide intuition, the analysis is organized around the following triangle inequality:

$$\|\mathbf{T}(\mathbf{q}) - \mathbf{y}^*(\mathbf{q})\|_2 \;\le\; \|\mathbf{T}(\mathbf{q}) - \mathbf{y}_{\tau,L}(\mathbf{q})\|_2 + \|\mathbf{y}_{\tau,L}(\mathbf{q}) - \mathbf{y}_\tau(\mathbf{q})\|_2 + \|\mathbf{y}_\tau(\mathbf{q}) - \mathbf{y}^*(\mathbf{q})\|_2, \quad (6)$$

[1] At 33×: Qwen - WIKI/LCC/SAMSUM/REPO use $(P, L) = (10, 70)$.

which isolates the contributions of sampling variance, finite-table effects, and soft-bucketization bias, respectively. Sections B.2 and B.3 revolve around bounding each of them separately.

Role of M and L. The terms $M^{-1/2}$ and $L^{-1/2}$ capture two independent variance sources. The $M^{-1/2}$ term arises from the value-aware sampling step and reflects the Monte Carlo error incurred when approximating the soft attention output using M sampled values. The $L^{-1/2}$ term corresponds to the finite-table approximation of the soft collision kernel and quantifies the concentration of the L-table estimator around its population limit. Both terms decay at the standard parametric rate and are independent of the sequence length N, highlighting the scalability of the method. They can be controlled separately depending on computational and memory constraints.

Role of the temperature τ. The term $\varepsilon_\tau(\mathbf{q})$ in Theorem 3 captures the approximation bias induced by soft bucketization. Fix a table ℓ and define the hard SRP bucket of the query by

$$b_q^{(\ell)} := h^{(\ell)}(\mathbf{q}) = \operatorname{sign}(\mathbf{W}^{(\ell)}\mathbf{q}) \in \{\pm 1\}^P$$

Recall that Algorithm 2 defines

$$p_\tau^{(\ell)}(r \mid \mathbf{q}) = \frac{\exp\big(\mathbf{u}^{(\ell)}(\mathbf{q})^\top \mathbf{c}_r/\tau\big)}{\sum_{r'=1}^R \exp\big(\mathbf{u}^{(\ell)}(\mathbf{q})^\top \mathbf{c}_{r'}/\tau\big)},$$

We define the peaking (non-dominant mass) quantity

$$\varepsilon_\tau(\mathbf{q}) := \mathbb{E}_{\mathbf{W}^{(\ell)}}\Big[\,1 - p_\tau^{(\ell)}\big(b_q^{(\ell)} \mid \mathbf{q}\big)\Big],$$

which measures how much probability mass the soft bucket distribution assigns to buckets other than the hard query bucket. To see how τ controls this bias, write the bucket logits as $x_r := \mathbf{u}^{(\ell)}(\mathbf{q})^\top \mathbf{c}_r$ for $r \in [R]$, so that $p_\tau^{(\ell)}(r \mid \mathbf{q}) = \exp(x_r/\tau)/\sum_{r'} \exp(x_{r'}/\tau)$. Let $r^\star := \arg\max_{r\in[R]} x_r$. Since $\tanh(\cdot)$ is strictly increasing coordinatewise, we have $r^\star = b_q^{(\ell)}$, and factoring out the maximum logit yields

$$p_\tau^{(\ell)}(b_q \mid \mathbf{q}) = \frac{\exp\big(x_{b_\mathbf{q}}/\tau\big)}{\sum_{r=1}^R \exp\big(x_r/\tau\big)} = \frac{1}{1 + \sum_{r\neq b_q} \exp\big((x_r - x_{b_q})/\tau\big)}$$

Because $x_r - x_{b_q} < 0$ for all $r \neq b_q$ and $R = 2^P$ is finite for fixed P, we have $\exp((x_r - x_{b_q})/\tau) \to 0$ as $\tau \to 0$, hence $p_\tau^{(\ell)}(b_q \mid \mathbf{q}) \to 1$ and therefore $\varepsilon_\tau(\mathbf{q}) \to 0$ as $\tau \to 0$ (for fixed P). In the opposite extreme $\tau \to \infty$, we have $\exp(x_r/\tau) = 1 + \mathcal{O}(1/\tau)$ and thus $p_\tau^{(\ell)}(r \mid \mathbf{q}) \to 1/R$ for all r, i.e., the bucket distribution uniformizes and $\varepsilon_\tau(\mathbf{q}) \to 1 - 1/R$. Consequently, τ interpolates between hard bucketing ($\tau \to 0$) and uniform bucket weights ($\tau \to \infty$), trading approximation bias against smoother, more stable similarity scores at finite τ.

Error-bound without sampling. If the value-aware sampling step is omitted, the estimator reduces to the finite-table soft-count attention output $\mathbf{y}_{\tau,L}(\mathbf{q})$. In this setting, the randomness arises solely from the L random hash tables, and the sampling variance term in Theorem 3 disappears. Consequently, under the same assumptions and with probability at least $1 - \delta$ over the hash-table randomness, we have

$$\|\mathbf{y}_{\tau,L}(\mathbf{q}) - \mathbf{y}^*(\mathbf{q})\|_2 = \widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{L}} + \varepsilon_\tau(\mathbf{q})\right)\|\mathbf{V}\|_2.$$

The $L^{-1/2}$ term captures the concentration of the finite-table soft-count estimator around its population limit $\mathbf{y}_\tau(\mathbf{q})$, while $\varepsilon_\tau(\mathbf{q})$ quantifies the bias introduced by soft bucketization relative to angular attention. Thus, in the absence of sampling, soft collision aggregation provides a statistically controlled proxy for angular attention, with accuracy governed by the number of hash tables and the temperature parameter.

B.2 Intermediate Lemmas

Fix a query $\mathbf{q} \in \mathbb{R}^d$, keys $\mathbf{k}_1, \dots \mathbf{k}_N \in \mathbb{R}^d$ and values $\mathbf{v}_1, \dots \mathbf{v}_N \in \mathbb{R}^d$. Define the angular kernel weights $w_j = \left(1 - \frac{1}{\pi}\cos^{-1}\left(\mathbf{q}^\top \mathbf{k}_j / \|\mathbf{q}\|\|\mathbf{k}_j\|\right)\right)^P \in [0,1]$ and $Z := \sum_{j=1}^N w_j$. With this, define the angular attention distribution and output:

$$a_j := \frac{w_j}{Z}, \quad \mathbf{y}^*(\mathbf{q}) := \sum_{j=1}^N a_j \boldsymbol{b}_j.$$

***Soft-count* distribution**: For each table $\ell \in [L]$, draw hyperplanes $\mathbf{W}^{(\ell)} \in \mathbb{R}^{P\times d}$ with i.i.d. $\mathcal{N}(0,1)$ rows. Let corners $\mathbf{c}_r \in \{\pm 1\}^P$ for $r \in [R]$, $R = 2^P$. Define

$$\mathbf{u}^{(\ell)}(\mathbf{q}) := \frac{1}{\sqrt{d}} \tanh\left(\mathbf{W}^{(\ell)}\mathbf{q}\right) \in \mathbb{R}^P, \quad p_\tau^{(\ell)}(r \mid \mathbf{q}) := \frac{\exp\left(\mathbf{u}^{(\ell)}(\mathbf{q})^\top \mathbf{c}_r/\tau\right)}{\sum_{r'=1}^R \exp\left(\mathbf{u}^{(\ell)}(\mathbf{q})^\top \mathbf{c}_{r'}/\tau\right)}.$$

Let the $\mathbf{k}_j$'s (hard) bucket id be $b_j^{(\ell)} \in [R]$ (equivalently the sign pattern of $\mathbf{W}^{(\ell)}\mathbf{k}_j$). Define the per-table soft score and the L-table soft-count:

$$s_j^{(\ell)}(\mathbf{q}) := p_\tau^{(\ell)}\left(b_j^{(\ell)} \mid \mathbf{q}\right) \in [0,1], \quad \widetilde{w}_j := \frac{1}{L}\sum_{\ell=1}^L s_j^{(\ell)}(\mathbf{q}), \quad \widetilde{Z} := \sum_{j=1}^N \widetilde{w}_j, \quad \widetilde{a}_j := \frac{\widetilde{w}_j}{\widetilde{Z}}.$$

Now, we define the "soft attention" output: $\mathbf{y}_{\tau,L}(\mathbf{q}) := \sum_{j=1}^N \widetilde{a}_j \boldsymbol{b}_j = \frac{\widetilde{n}(\mathbf{q})}{\widetilde{Z}} \in \mathbb{R}^d$. Similarly, we define the population (single-table) soft-count weight $w_{\tau,j} := \mathbb{E}\left[s_j^{(1)}(\mathbf{q})\right]$, $Z_\tau := \sum_{j=1}^N w_{\tau,j}$, $a_{\tau,j} := \frac{w_{\tau,j}}{Z_\tau}$. Accordingly, the soft-attention output of the population is given by $\mathbf{y}_\tau(\mathbf{q}) := \sum_{j=1}^N a_{\tau,j}\boldsymbol{b}_j = \frac{\mathbf{n}_\tau(\mathbf{q})}{Z_\tau}$, where the expectations are over the randomness in the table, i.e. $\mathbf{W}^{(\ell)}$ and $\mathbf{n}_\tau(\mathbf{q}) = \sum_{j=1}^N w_{\tau,j}\boldsymbol{b}_j \in \mathbb{R}^d$.

Note that Assumption 2 directly implies

$$\begin{aligned} Z^{(\ell)}(\mathbf{q}) := \sum_{j=1}^N s_j^{(\ell)}(\mathbf{q}) &= \sum_{j=1}^N p_\tau^{(\ell)}\left(b_j^{(\ell)} \mid \mathbf{q}\right) = \sum_{r=1}^R \sum_{b_j^{(\ell)}=r} p_\tau^{(\ell)}(r \mid \mathbf{q}) \\ &= \sum_{r=1}^R p_\tau^{(\ell)}(r \mid \mathbf{q})\Big(\sum_{b_j^{(\ell)}=r} 1\Big) \le B\sum_{r=1}^R p_\tau^{(\ell)}(r \mid \mathbf{q}) = B. \end{aligned} \tag{7}$$

Lemma 5 (Concentration of $\widetilde{Z}$ and $\widetilde{\mathbf{n}}(\mathbf{q})$) *If $L \ge \frac{2B^2\log(4/\delta_{\mathrm{L}})}{Z_{\tau,\min}^2}$, then with probability at least $1-\delta_{\mathrm{L}}$,*

$$\left|\widetilde{Z} - Z_\tau\right| \le B\sqrt{\frac{\log(4/\delta_{\mathrm{L}})}{2L}} \quad \textit{and} \quad \|\widetilde{\mathbf{n}} - \mathbf{n}_\tau\|_2 \le 2\|\mathbf{V}\|_2\sqrt{2B}\sqrt{\frac{\log(4/\delta_{\mathrm{L}})}{L}}, \tag{8}$$

and moreover $\widetilde{Z} \ge \frac{Z_{\tau,\min}}{2}$.

Proof Let $\mathbf{V} \in \mathbb{R}^{N\times d}$ be the value matrix such that $\mathbf{V}^\top := [\mathbf{v}_1, \dots \mathbf{v}_N] \in \mathbb{R}^{d\times N}$. For each table $\ell \in [L]$, define the score vector $\mathbf{s}^{(\ell)}(\mathbf{q}) := (s_1^{(\ell)}(\mathbf{q}), \dots, s_N^{(\ell)}(\mathbf{q}))^\top \in \mathbb{R}^N$ and $\|\mathbf{V}\|_2$ as the spectral norm of $\mathbf{V}$. From eq. equation 7, we already know $0 \le Z^{(\ell)}(\mathbf{q}) \le B$ for every $\ell \in [L]$. Next, we define the per-table numerator vector $\mathbf{n}^{(\ell)}(\mathbf{q}) := \sum_{j=1}^N s_j^{(\ell)}(\mathbf{q})\boldsymbol{b}_j = \mathbf{V}^\top \mathbf{s}^{(\ell)}(\mathbf{q}) \in \mathbb{R}^d$. From the

notations we already defined in the beginning of Section B.2, we have

$$\widetilde{\mathbf{n}}(\mathbf{q}) = \sum_{j=1}^{N} \widetilde{w}_j \boldsymbol{b}_j = \frac{1}{L} \sum_{\ell=1}^{L} \sum_{j=1}^{N} s_j^{(\ell)}(\mathbf{q}) \boldsymbol{b}_j = \frac{1}{L} \sum_{\ell=1}^{L} \mathbf{n}^{(\ell)}(\mathbf{q}) \in \mathbb{R}^d \tag{9}$$

$$\widetilde{Z} = \sum_{j=1}^{N} \widetilde{w}_j = \sum_{j=1}^{N} \left(\frac{1}{L} \sum_{\ell=1}^{L} s_j^{(\ell)}(\mathbf{q}) \right) = \frac{1}{L} \sum_{\ell=1}^{L} Z^{(\ell)}(\mathbf{q}). \tag{10}$$

Bounding $\|\mathbf{n}^{(\ell)}(\mathbf{q})\|_2$: Since $0 \le s_j^{(\ell)}(\mathbf{q}) \le 1$ for all $j \in [N]$, we have

$$\|\mathbf{s}^{(\ell)}(\mathbf{q})\|_2^2 = \sum_{j=1}^{N} (s_j^{(\ell)}(\mathbf{q}))^2 \le \sum_{j=1}^{N} s_j^{(\ell)}(\mathbf{q}) = Z^{(\ell)}(\mathbf{q}) \le B. \tag{11}$$

Now, combining eq. equation 11 and submultiplicativity of the spectral norm, we have

$$\|\mathbf{n}^{(\ell)}(\mathbf{q})\|_2 = \|\mathbf{V}^{\mathsf{T}} \mathbf{s}^{(\ell)}(\mathbf{q})\|_2 \le \|\mathbf{V}\|_2 \|\mathbf{s}^{(\ell)}(\mathbf{q})\|_2 \le \|\mathbf{V}\|_2 \sqrt{B}. \tag{12}$$

Bounding $\widetilde{Z}$: From eq. equation 7, each $Z^{(\ell)}(\mathbf{q}) \in [0, B]$, and the $Z^{(\ell)}$ are i.i.d. across ℓ with

$$\begin{aligned} \mathbb{E}(\widetilde{Z}) =& \mathbb{E}\left(\frac{1}{L} \sum_{\ell=1}^{L} Z^{(\ell)}(\mathbf{q}) \right) = \mathbb{E}(Z^{(\ell)}(\mathbf{q})) \\ =& \mathbb{E}\left(\sum_{j=1}^{N} s_j^{(\ell)}(\mathbf{q}) \right) = \sum_{j=1}^{N} \mathbb{E}\left(s_j^{(\ell)}(\mathbf{q}) \right) \\ =& \sum_{j=1}^{N} w_{\tau,j} = Z_\tau. \end{aligned} \tag{13}$$

Therefore, applying Hoeffding inequality with $t > 0$, we get $\mathbb{P}\left(|\widetilde{Z} - Z_\tau| \ge t \right) \le 2 \exp\left(-\frac{2Lt^2}{B^2} \right)$. Taking $t = B\sqrt{\frac{\log(4/\delta_{\mathrm{L}})}{2L}}$, we have

$$\mathbb{P}\left(|\widetilde{Z} - Z_\tau| \le B\sqrt{\frac{\log(4/\delta_{\mathrm{L}})}{2L}} \right) \ge 1 - \delta_{\mathrm{L}}/2. \tag{14}$$

Now, if $B\sqrt{\frac{\log(4/\delta_{\mathrm{L}})}{2L}} \le \frac{Z_{\tau,\min}}{2}$ or $L \ge \frac{2B^2 \log(4/\delta_{\mathrm{L}})}{Z_{\tau,\min}^2}$ then combining Assumption 1 with eq. equation 14 further boils down to

$$\mathbb{P}\left(\widetilde{Z} \ge \frac{Z_{\tau,\min}}{2} \right) \ge 1 - \delta_{\mathrm{L}}/2. \tag{15}$$

Bounding $\widetilde{\mathbf{n}}(\mathbf{q})$: Let $\mathbf{X}_\ell := \mathbf{n}^{(\ell)}(\mathbf{q}) - \mathbf{n}_\tau(\mathbf{q}) \in \mathbb{R}^d$. Then $\mathbb{E}[\mathbf{X}_\ell] = 0$ and the $\mathbf{X}_\ell$ are i.i.d. Using eq. equation 12 and Jensen's inequality, we have

$$\|\mathbf{n}_\tau(\mathbf{q})\|_2 = \|\mathbb{E}(\mathbf{n}^{(\ell)}(\mathbf{q}))\|_2 \le \mathbb{E}\|\mathbf{n}^{(\ell)}(\mathbf{q})\|_2 \le \|\mathbf{V}\|_2 \sqrt{B}. \tag{16}$$

So, applying triangle inequality on the definition of $\mathbf{X}_\ell$ and combining eqs. equation 12 & equation 16 yield

$$\|\mathbf{X}_\ell\|_2 \le \|\mathbf{n}^{(\ell)}(\mathbf{q})\|_2 + \|\mathbf{n}_\tau(\mathbf{q})\|_2 \le 2\|\mathbf{V}\|_2 \sqrt{B}. \tag{17}$$

Next, we apply a vector Hoeffding inequality for sums of bounded mean-zero random vectors in $\mathbb{R}^d$. Specifically, if $\mathbf{X}_1, \ldots, \mathbf{X}_K$ are independent with $\mathbb{E}[\mathbf{X}_k] = 0$ and $\|\mathbf{X}_k\|_2 \le R$, then

$$\mathbb{P}\left(\left\| \sum_{k=1}^{K} \mathbf{X}_k \right\|_2 \ge t \right) \le 2 \exp\left(-\frac{t^2}{2R^2} \right),$$

which follows from *e.g.,* Theorem 3 in Pinelis (1992) and related vector concentration results (e.g., Boucheron et al. (2013)). Applying this in our context directly yields

$$\mathbb{P}\left(\|\widetilde{\mathbf{n}} - \mathbf{n}_\tau\|_2 \geq t\right) \leq 2\exp\left(-\frac{Lt^2}{8\|\mathbf{V}\|_2^2 B}\right).$$

Setting $t = 2\|\mathbf{V}\|_2\sqrt{2B}\sqrt{\frac{\log(4/\delta_{\mathrm{L}})}{L}}$ above yields:

$$\mathbb{P}\left(\|\widetilde{\mathbf{n}} - \mathbf{n}_\tau\|_2 \leq 2\|\mathbf{V}\|_2\sqrt{2B}\sqrt{\frac{\log(4/\delta_{\mathrm{L}})}{L}}\right) \geq 1 - \delta_{\mathrm{L}}/2. \tag{18}$$

Finally, taking the union bound over eqs. equation 14 and equation 18 gives the stated joint event with probability at least $1 - \delta_{\mathrm{L}}$, and under $L \geq \frac{2B^2\log(4/\delta_{\mathrm{L}})}{Z_{\tau,\min}^2}$ we also have $\widetilde{Z} \geq \frac{Z_{\tau,\min}}{2}$ by eq. equation 15. ■

Lemma 6 (Final bound for $\mathbf{y}_{\tau,L}(\mathbf{q})$) *If $L \geq \frac{2B^2\log(4/\delta_{\mathrm{L}})}{Z_{\tau,\min}^2}$, then*

$$\mathbb{P}\Big(\|\mathbf{y}_{\tau,L}(\mathbf{q}) - \mathbf{y}_\tau(\mathbf{q})\|_2 \leq C\sqrt{\frac{\log(4/\delta_{\mathrm{L}})}{L}}\|\mathbf{V}\|_2\Big) \geq 1 - \delta_{\mathrm{L}},$$

where $C = \left(\frac{4\sqrt{2B}}{Z_{\tau,\min}} + \frac{\sqrt{2}B^{3/2}}{Z_{\tau,\min}^2}\right)$.

Proof Using the definitions of $\mathbf{y}_{\tau,L}(\mathbf{q})$ and $\mathbf{y}_\tau(\mathbf{q})$, and adding and subtracting $\mathbf{n}_\tau/\widetilde{Z}$, we have

$$\mathbf{y}_{\tau,L}(\mathbf{q}) - \mathbf{y}_\tau(\mathbf{q}) = \frac{\widetilde{\mathbf{n}}}{\widetilde{Z}} - \frac{\mathbf{n}_\tau}{Z_\tau} = \frac{\widetilde{\mathbf{n}} - \mathbf{n}_\tau}{\widetilde{Z}} + \mathbf{n}_\tau\left(\frac{1}{\widetilde{Z}} - \frac{1}{Z_\tau}\right). \tag{19}$$

Taking norms both sides and applying triangle inequality yield

$$\|\mathbf{y}_{\tau,L}(\mathbf{q}) - \mathbf{y}_\tau(\mathbf{q})\|_2 \leq \frac{\|\widetilde{\mathbf{n}} - \mathbf{n}_\tau\|_2}{\widetilde{Z}} + \frac{\|\mathbf{n}_\tau\|_2}{\widetilde{Z}Z_\tau}|\widetilde{Z} - Z_\tau|. \tag{20}$$

Now using Assumption 1 *i.e.,* $Z_\tau \geq Z_{\tau,\min}$, $\widetilde{Z} \geq Z_{\tau,\min}/2$ from Lemma 5, and $\|\mathbf{n}_\tau\|_2 \leq \|\mathbf{V}\|_2\sqrt{B}$ in eq. equation 16:

$$\|\mathbf{y}_{\tau,L}(\mathbf{q}) - \mathbf{y}_\tau(\mathbf{q})\|_2 \leq \frac{2}{Z_{\tau,\min}}\|\widetilde{\mathbf{n}} - \mathbf{n}_\tau\|_2 + \frac{2\|\mathbf{V}\|_2\sqrt{B}}{Z_{\tau,\min}^2}\left|\widetilde{Z} - Z_\tau\right|. \tag{21}$$

Finally, applying the bounds

$$\|\widetilde{\mathbf{n}} - \mathbf{n}_\tau\|_2 \leq 2\|\mathbf{V}\|_2\sqrt{2B}\sqrt{\frac{\log(4/\delta_{\mathrm{L}})}{L}} \quad \text{and} \quad \left|\widetilde{Z} - Z_\tau\right| \leq B\sqrt{\frac{\log(4/\delta_{\mathrm{L}})}{2L}}$$

(which hold jointly with probability at least $1 - \delta_{\mathrm{L}}$ by Lemma 5), we get

$$\begin{aligned}\|\mathbf{y}_{\tau,L}(\mathbf{q}) - \mathbf{y}_\tau(\mathbf{q})\|_2 &\leq \frac{2}{Z_{\tau,\min}}\left(2\|\mathbf{V}\|_2\sqrt{2B}\sqrt{\frac{\log(4/\delta_{\mathrm{L}})}{L}}\right) + \frac{2\|\mathbf{V}\|_2\sqrt{B}}{Z_{\tau,\min}^2}\left(B\sqrt{\frac{\log(4/\delta_{\mathrm{L}})}{2L}}\right) \\ &= \left(\frac{4\sqrt{2B}}{Z_{\tau,\min}} + \frac{\sqrt{2}B^{3/2}}{Z_{\tau,\min}^2}\right)\sqrt{\frac{\log(4/\delta_{\mathrm{L}})}{L}}\|\mathbf{V}\|_2. \end{aligned} \tag{22}$$

This concludes the proof. ■

Conditioned on the tables, define $S_1 := \sum_{j=1}^N \widetilde{a}_j\|\boldsymbol{b}_j\|_2$ and sampling probabilities $p_j := \widetilde{a}_j\|\boldsymbol{b}_j\|_2/S_1$. Draw i.i.d. indices $J_1, \ldots, J_M \sim p$ and define

$$\mathbf{T}(\mathbf{q}) := \frac{1}{M}\sum_{m=1}^M \frac{\widetilde{a}_{J_m}}{p_{J_m}}\mathbf{v}_{J_m} \in \mathbb{R}^d.$$

Let $\mathcal{W} = \{\mathbf{W}_1, \ldots, \mathbf{W}_L\}$ be the set of random Gaussian hyperplanes with $\mathbf{W}_\ell \in \mathbb{R}^{P\times d}$. Now are ready to present the results for sampling.

Lemma 7 *The following properties hold:*

1. $\mathbb{E}[\mathbf{T}(\mathbf{q}) \mid \mathcal{W}] = \mathbf{y}_{\tau,L}(\mathbf{q})$
2. $\mathbb{P}\left(\|\mathbf{T}(\mathbf{q}) - \mathbf{y}_{\tau,L}(\mathbf{q})\|_2 \leq \|\mathbf{V}\|_2 \sqrt{\frac{8\log(2/\delta_{\mathrm{M}})}{M}}\right) \geq 1 - \delta_{\mathrm{M}}$

Proof

Proof of Part 1. Let $J_1, \ldots, J_M$ be the M independent samples of $\{1, \ldots, N\}$ drawn using the probability distribution $p := (p_1, \ldots, p_N)$. Define $\mathbf{X}_m := \frac{\widetilde{a}_{J_m}}{p_{J_m}} \mathbf{v}_{J_m} \in \mathbb{R}^d$. Then

$$\mathbf{T}(\mathbf{q}) = \frac{1}{M} \sum_{m=1}^{M} \mathbf{X}_m$$

Now, given $\mathcal{W}$. Then $\widetilde{a}_j$ and p_j are fixed numbers, so:

$$\mathbb{E}[\mathbf{X}_m \mid \mathcal{W}] = \sum_{j=1}^{N} p_j \frac{\widetilde{a}_j}{p_j} \boldsymbol{b}_j = \sum_{j=1}^{N} \widetilde{a}_j \boldsymbol{b}_j = \mathbf{y}_{\tau,L}(\mathbf{q}).$$

Therefore,

$$\mathbb{E}[\mathbf{T}(\mathbf{q}) \mid \mathcal{W}] = \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}\left[\mathbf{X}_m \mid \mathcal{W}\right] = \mathbf{y}_{\tau,L}(\mathbf{q})$$

Proof of Part 2. We define the centered random vector $\mathbf{Y}_m := \mathbf{X}_m - \mathbb{E}\left[\mathbf{X}_m \mid \mathcal{W}\right] = \mathbf{X}_m - \mathbf{y}_{\tau,L}(\mathbf{q})$. Then $\mathbb{E}\left[\mathbf{Y}_m \mid \mathcal{W}\right] = 0$ and $\mathbf{T}(\mathbf{q}) - \mathbf{y}_{\tau,L}(\mathbf{q}) = \frac{1}{M} \sum_{m=1}^{M} \mathbf{Y}_m$. Now, using $p_{J_m} := \frac{\widetilde{a}_{J_m} \|\mathbf{v}_{J_m}\|_2}{S_1}$, we have

$$\|\mathbf{X}_m\|_2 = \left\| \frac{\widetilde{a}_{J_m}}{p_{J_m}} \mathbf{v}_{J_m} \right\|_2 = \frac{\widetilde{a}_{J_m}}{\widetilde{a}_{J_m} \|\mathbf{v}_{J_m}\|_2 / S_1} \|\mathbf{v}_{J_m}\|_2 = S_1 \tag{23}$$

and also applying triangle inequality

$$\|\mathbf{y}_{\tau,L}(\mathbf{q})\|_2 = \left\| \sum_{j=1}^{N} \widetilde{a}_j \boldsymbol{b}_j \right\|_2 \leq \sum_{j=1}^{N} \widetilde{a}_j \|\boldsymbol{b}_j\|_2 = S_1. \tag{24}$$

Therefore, applying the triangle inequality on $\mathbf{Y}_m$, we have

$$\|\mathbf{Y}_m\|_2 \leq \|\mathbf{X}_m\|_2 + \|\mathbf{y}_{\tau,L}(\mathbf{q})\|_2 = 2S_1 \tag{25}$$

Now, note that we can always rewrite $\boldsymbol{b}_j = \mathbf{V}^{\mathsf{T}} \mathbf{e}_j$ where $\mathbf{e}_j \in \mathbb{R}N$ is the j-th standard basis vector and thus $\|\mathbf{e}_j\|_2 = 1$. Consequently, applying the submultiplicativity property, we trivially have

$$\|\boldsymbol{b}_j\|_2 = \|\mathbf{V}^{\mathsf{T}} \mathbf{e}_j\|_2 \leq \|\mathbf{e}_j\|_2 \|\mathbf{V}\|_2 = \|\mathbf{V}\|_2 \tag{26}$$

Using the definition of S_1, eq. equation 26 and the fact that $\sum_{j=1}^{N} \widetilde{a}_j$=1, we get

$$S_1 = \sum_{j=1}^{N} \widetilde{a}_j \|\boldsymbol{b}_j\|_2 \leq \|\mathbf{V}\|_2 \tag{27}$$

Therefore $\|\mathbf{Y}_m\|_2 \leq 2\|\mathbf{V}\|_2$. Now, using the fact that $\mathbf{T}(\mathbf{q}) - \mathbf{y}_{\tau,L}(\mathbf{q}) = \frac{1}{M} \sum_{m=1}^{M} \mathbf{Y}_m$ and applying the same vector Hoeffding bound we have

$$\mathbb{P}\left(\|\mathbf{T}(\mathbf{q}) - \mathbf{y}_{\tau,L}(\mathbf{q})\|_2 \geq t \mid \mathcal{W}\right) = \mathbb{P}\left(\left\|\frac{1}{M}\sum_{m=1}^{M}\mathbf{Y}_m\right\|_2 \geq t \mid \mathcal{W}\right) \leq 2\exp\left(-\frac{Mt^2}{8\|\mathbf{V}\|_2^2}\right) \quad (28)$$

Now, for a failure probability $0 < \delta_{\mathrm{M}} < 1$, we take $t = \sqrt{\frac{8\log(2/\delta_{\mathrm{M}})}{M}}\|\mathbf{V}\|_2$ and eq. equation 28 further simplify to

$$\mathbb{P}\left(\|\mathbf{T}(\mathbf{q}) - \mathbf{y}_{\tau,L}(\mathbf{q})\|_2 \leq \sqrt{\frac{8\log(2/\delta_{\mathrm{M}})}{M}}\|\mathbf{V}\|_2 \mid \mathcal{W}\right) \geq 1 - \delta_{\mathrm{M}} \quad (29)$$

Finally, since eq. equation 29 holds for any $\mathcal{W}$, we take the expectations over both sides with respect to $\mathcal{W}$ and use the law of iterated expectation to get

$$\mathbb{P}\left(\left\|\mathbf{T}(\mathbf{q}) - \mathbf{y}_{\tau,L}(\mathbf{q})\right\|_2 \leq \sqrt{8\log(2/\delta_{\mathrm{M}})/M}\|\mathbf{V}\|_2\right) \geq 1 - \delta_{\mathrm{M}} \quad (30)$$

■

Lemma 8 *For any fixed query* $\mathbf{q}$*, we have*

$$\|\mathbf{y}_\tau(\mathbf{q}) - \mathbf{y}^*(\mathbf{q})\|_2 \leq 2B\left(\frac{1}{Z_{\tau,\min}} + \frac{\sqrt{B}}{Z_{\min}Z_{\tau,\min}}\right)\varepsilon_\tau(\mathbf{q})\,\|\mathbf{V}\|_2.$$

Proof From the definition of $\mathbf{y}_\tau(\mathbf{q})$ and $\mathbf{y}^*(\mathbf{q})$ we have

$$\|\mathbf{y}_\tau(\mathbf{q}) - \mathbf{y}^*(\mathbf{q})\|_2 = \Big\|\sum_{j=1}^{N}(a_{\tau,j} - a_j)\boldsymbol{b}_j\Big\|_2 = \left\|\mathbf{V}^\mathsf{T}\,(_\tau -)\right\|_2 \leq \|\mathbf{V}\|_2\|_\tau - \|_2\,, \quad (31)$$

where $_\tau$ and are the vectors of order N whose j-th entries are $a_{\tau,j}$ and a_j respectively. We applied the submultiplicativity property of vector and matrix norms in the last inequality of eq. equation 31. So it remains to bound $\|_\tau - \|_2$. Now, from the definitions of $a_{\tau,j}$ and a_j in the beginning of Section B.2, we rewrite $(_\tau -)$ as

$$_\tau - = \frac{\mathbf{w}_\tau}{Z_\tau} - \frac{\mathbf{w}}{Z} = \frac{\mathbf{w}_\tau - \mathbf{w}}{Z_\tau} + \frac{Z - Z_\tau}{Z_\tau Z}\,\mathbf{w}\,, \quad (32)$$

where $\mathbf{w}_\tau$ and $\mathbf{w}$ are the unnormalized attention vectors of dimension N whose j-th entries are $w_{\tau,j}$ and w_j respectively. The last equality follows from adding and subtracting $\mathbf{w}/Z_\tau$ from the second equality and then simplifying. Now, taking ℓ_2-norm on the both sides of eq. equation 32 and applying triangle inequality, we further have

$$\begin{aligned}\|_\tau - \|_2 \leq& \frac{\|\mathbf{w}_\tau - \mathbf{w}\|_2}{Z_\tau} + \frac{|Z - Z_\tau|}{Z_\tau Z}\,\|\mathbf{w}\|_2 \\ \leq& \frac{\|\mathbf{w}_\tau - \mathbf{w}\|_2}{Z_{\tau,\min}} + \frac{|Z - Z_\tau|}{Z_{\tau,\min}\,Z_{\min}}\,\|\mathbf{w}\|_2\,, \end{aligned} \quad (33)$$

where the last inequality in eq. equation 33 follows directly from Assumption 1.

Now, we rewrite w_j as an SRP collision probability. Fix a single table with $\mathbf{W}^{(\ell)} \sim \mathcal{N}(0,1)^{P\times d}$. Define the hard hash (bucket id) of any $\mathbf{x} \in \mathbb{R}d$ by

$$h(\mathbf{x}) := \mathrm{sign}(\mathbf{W}^{(\ell)}\mathbf{x}) \in \{\pm 1\}^P \quad \text{(equivalently an index in } [R]).$$

Then $b_j^{(\ell)}$ is precisely the index of $h(\mathbf{k}_j)$. Let $b_q := h(\mathbf{q})$ denote the query bucket under the same table. Define the hard collision indicator $I_j := \mathbf{1}\{b_j^{(\ell)} = b_q\} \in \{0,1\}$. By the standard SRP/SimHash collision identity, we rewrite

$$w_j = \left(1 - \frac{1}{\pi}\cos^{-1}\left(\frac{\mathbf{q}^\top\mathbf{k}_j}{\|\mathbf{q}\|\,\|\mathbf{k}_j\|}\right)\right)^P = \mathbb{E}[I_j],$$

where the expectation is over $\mathbf{W}^{(\ell)}$. We now express $w_{\tau,j}$ as the soft score expectation and define the peaking quantity. Recall $w_{\tau,j} := \mathbb{E}[s_j^{(\ell)}(\mathbf{q})]$ where $s_j^{(\ell)}(\mathbf{q}) = p_\tau^{(\ell)}(b_j^{(\ell)} \mid \mathbf{q}) \in [0,1]$. For the same table, define the (random) dominant query bucket

$$b_\star := \arg\max_{r\in[R]} p_\tau^{(\ell)}(r \mid \mathbf{q}).$$

Since $p_\tau^{(\ell)}(r \mid \mathbf{q}) \propto \exp(\mathbf{u}^{(\ell)}(\mathbf{q})^\top \mathbf{c}_r/\tau)$, $\mathbf{u}^{(\ell)}(\mathbf{q}) = \frac{1}{\sqrt{d}}\tanh(\mathbf{W}^{(\ell)}\mathbf{q})$, and $\tanh(\cdot)$ is strictly increasing, we have $\operatorname{sign}\big(\mathbf{u}^{(\ell)}(\mathbf{q})\big) = \operatorname{sign}\big(\mathbf{W}^{(\ell)}\mathbf{q}\big)$ coordinatewise. Hence the dominant query bucket under soft collision and the hard bucket for $\mathbf{q}$ coincides *i.e.*, $b_\star = b_q$. Now, we define the non-dominant mass quantity

$$\varepsilon_\tau(\mathbf{q}) := \mathbb{E}\big[1 - p_\tau^{(\ell)}(b_q \mid \mathbf{q})\big] = \mathbb{E}\big[1 - p_\tau^{(\ell)}(b_\star \mid \mathbf{q})\big],$$

where the expectation is over the randomness of the table $\mathbf{W}^{(\ell)}$. In fact, $\varepsilon_\tau(\mathbf{q})$ quantifies how much probability mass the soft bucket distribution assigns to buckets other than the hard SRP bucket, and it controls the discrepancy between soft-count weights and hard collision probabilities.

Now, we are ready to bound $\|\mathbf{w}_\tau - \mathbf{w}\|_1$ and $|Z - Z_\tau|$ using occupancy B. We first prove a *per-table* inequality. For a fixed realization of $\mathbf{W}^{(\ell)}$, we group indices by buckets and use Assumption 2. Write

$$\sum_{j=1}^{N} |s_j^{(\ell)}(\mathbf{q}) - I_j| = \sum_{b_j^{(\ell)}=b_q} \big|p_\tau^{(\ell)}(b_q \mid \mathbf{q}) - 1\big| + \sum_{b_j^{(\ell)}\neq b_q} p_\tau^{(\ell)}(b_j^{(\ell)} \mid \mathbf{q}). \tag{34}$$

For the first term, there are at most B indices in bucket b_q, hence

$$\sum_{b_j^{(\ell)}=b_q} \big|p_\tau^{(\ell)}(b_q \mid \mathbf{q}) - 1\big| = \Big(\sum_{b_j^{(\ell)}=b_q} 1\Big)\big(1 - p_\tau^{(\ell)}(b_q \mid \mathbf{q})\big) \le B\big(1 - p_\tau^{(\ell)}(b_q \mid \mathbf{q})\big). \tag{35}$$

For the second term, we regroup by buckets:

$$\sum_{b_j^{(\ell)}\neq b_q} p_\tau^{(\ell)}(b_j^{(\ell)} \mid \mathbf{q}) = \sum_{r\neq b_q} \sum_{b_j^{(\ell)}=r} p_\tau^{(\ell)}(r \mid \mathbf{q}) \le \sum_{r\neq b_q} B\, p_\tau^{(\ell)}(r \mid \mathbf{q}) = B\big(1 - p_\tau^{(\ell)}(b_q \mid \mathbf{q})\big). \tag{36}$$

In the above, the first equality in eq. equation 36 utilizes the fact that each bucket $r \in [R]$, all keys j with $b_j^{(\ell)} = r$ contribute the same probability value $p_\tau^{(\ell)}(r \mid \mathbf{q})$ (because the probability depends only on the bucket r, not on j). Then, inside the inner sum, the term $p_\tau^{(\ell)}(r \mid \mathbf{q})$ is constant w.r.t. j and the inequality follows directly from Assumption 2.

Combining eqs. equation 34, equation 35 and equation 36 yield the deterministic bound

$$\sum_{j=1}^{N} |s_j^{(\ell)}(\mathbf{q}) - I_j| \le 2B\big(1 - p_\tau^{(\ell)}(b_q \mid \mathbf{q})\big). \tag{37}$$

Now take expectation over $\mathbf{W}^{(\ell)}$ and use $w_{\tau,j} = \mathbb{E}[s_j^{(\ell)}(\mathbf{q})]$ and $w_j = \mathbb{E}[I_j]$:

$$\begin{aligned}\|\mathbf{w}_\tau - \mathbf{w}\|_1 &= \sum_{j=1}^{N} |w_{\tau,j} - w_j| = \sum_{j=1}^{N} \big|\mathbb{E}[s_j^{(\ell)}(\mathbf{q}) - I_j]\big| \le \sum_{j=1}^{N} \mathbb{E}|s_j^{(\ell)}(\mathbf{q}) - I_j| \\ &= \mathbb{E}\Big[\sum_{j=1}^{N} |s_j^{(\ell)}(\mathbf{q}) - I_j|\Big] \le 2B\,\mathbb{E}\big[1 - p_\tau^{(\ell)}(b_q \mid \mathbf{q})\big] = 2B\,\varepsilon_\tau(\mathbf{q}).\end{aligned} \tag{38}$$

Since $\|\cdot\|_2 \le \|\cdot\|_1$, we immediately get

$$\|\mathbf{w}_\tau - \mathbf{w}\|_2 \le \|\mathbf{w}_\tau - \mathbf{w}\|_1 \le 2B\,\varepsilon_\tau(\mathbf{q}). \tag{39}$$

Next, note that $Z = \sum_{j=1}^{N} w_j$ and $Z_\tau = \sum_{j=1}^{N} w_{\tau,j}$, hence

$$|Z - Z_\tau| = \Big|\sum_{j=1}^{N}(w_j - w_{\tau,j})\Big| \le \sum_{j=1}^{N}|w_j - w_{\tau,j}| = \|\mathbf{w}_\tau - \mathbf{w}\|_1 \le 2B\,\varepsilon_\tau(\mathbf{q}). \tag{40}$$

Next, to bound $\|\mathbf{w}\|_2$ using occupancy B, we use that $w_j \in [0,1]$ and the following shows $Z \le B$. Indeed, for a fixed table,

$$\sum_{j=1}^{N} I_j = \#\{j \in [N] : b_j^{(\ell)} = b_q\} \le B \quad \text{by Assumption 2.}$$

Taking expectation over $\mathbf{W}^{(\ell)}$ gives

$$Z = \sum_{j=1}^{N} w_j = \sum_{j=1}^{N}\mathbb{E}[I_j] = \mathbb{E}\Big[\sum_{j=1}^{N} I_j\Big] \le B.$$

Moreover, since $0 \le w_j \le 1$,

$$\|\mathbf{w}\|_2^2 = \sum_{j=1}^{N} w_j^2 \le \sum_{j=1}^{N} w_j = Z \le B,$$

hence

$$\|\mathbf{w}\|_2 \le \sqrt{B}. \tag{41}$$

Next, we substitute eqs. equation 39, equation 40, and equation 41 into eq. equation 33:

$$\begin{aligned}\|_\tau - \|_2 &\le \frac{\|\mathbf{w}_\tau - \mathbf{w}\|_2}{Z_{\tau,\min}} + \frac{|Z - Z_\tau|}{Z_{\tau,\min}Z_{\min}}\,\|\mathbf{w}\|_2 \\ &\le \frac{2B\,\varepsilon_\tau(\mathbf{q})}{Z_{\tau,\min}} + \frac{2B\,\varepsilon_\tau(\mathbf{q})}{Z_{\tau,\min}Z_{\min}}\cdot\sqrt{B} \\ &= 2B\Big(\frac{1}{Z_{\tau,\min}} + \frac{\sqrt{B}}{Z_{\min}Z_{\tau,\min}}\Big)\varepsilon_\tau(\mathbf{q}).\end{aligned} \tag{42}$$

Finally, combining eq. equation 31 with eq. equation 42:

$$\|\mathbf{y}_\tau(\mathbf{q}) - \mathbf{y}^*(\mathbf{q})\|_2 \le \|\mathbf{V}\|_2 \cdot 2B\Big(\frac{1}{Z_{\tau,\min}} + \frac{\sqrt{B}}{Z_{\min}Z_{\tau,\min}}\Big)\varepsilon_\tau(\mathbf{q}). \tag{43}$$

This proves the lemma.

■

B.3 Proof of Theorem 3

Theorem 9 (Final end-to-end bound via union bound) *Fix a query* $\mathbf{q}$*. Suppose Assumptions 1–2 hold and let* $\delta_{\mathrm{L}}, \delta_{\mathrm{M}} \in (0,1)$*. Assume*

$$L \;\ge\; \frac{2B^2\log(4/\delta_{\mathrm{L}})}{Z_{\tau,\min}^2}.$$

Then, with probability at least $1 - (\delta_{\mathrm{L}} + \delta_{\mathrm{M}})$*,*

$$\begin{aligned}\|\mathbf{T}(\mathbf{q}) - \mathbf{y}^*(\mathbf{q})\|_2 \;\le\; &\underbrace{\|\mathbf{V}\|_2\sqrt{\frac{8\log(2/\delta_{\mathrm{M}})}{M}}}_{\textit{sampling error}} + \underbrace{C\,\|\mathbf{V}\|_2\sqrt{\frac{\log(4/\delta_{\mathrm{L}})}{L}}}_{\textit{finite-}L\textit{ error}} \\ &+ \underbrace{2B\Big(\frac{1}{Z_{\tau,\min}} + \frac{\sqrt{B}}{Z_{\min}Z_{\tau,\min}}\Big)\,\varepsilon_\tau(\mathbf{q})\,\|\mathbf{V}\|_2}_{\textit{soft vs. angular bias}},\end{aligned} \tag{44}$$

where

$$C = \Big(\frac{4\sqrt{2B}}{Z_{\tau,\min}} + \frac{\sqrt{2}B^{3/2}}{Z_{\tau,\min}^2}\Big), \qquad \varepsilon_\tau(\mathbf{q}) = \mathbb{E}\Big[1 - p_\tau^{(\ell)}(b_q \mid \mathbf{q})\Big].$$

Proof By the triangle inequality,

$$\|\mathbf{T}(\mathbf{q}) - \mathbf{y}^*(\mathbf{q})\|_2 \le \|\mathbf{T}(\mathbf{q}) - \mathbf{y}_{\tau,L}(\mathbf{q})\|_2 + \|\mathbf{y}_{\tau,L}(\mathbf{q}) - \mathbf{y}_\tau(\mathbf{q})\|_2 + \|\mathbf{y}_\tau(\mathbf{q}) - \mathbf{y}^*(\mathbf{q})\|_2. \quad (45)$$

Define the events

$$\mathcal{E}_{\mathrm{M}} := \Big\{\|\mathbf{T}(\mathbf{q}) - \mathbf{y}_{\tau,L}(\mathbf{q})\|_2 \le \|\mathbf{V}\|_2\sqrt{\tfrac{8\log(2/\delta_{\mathrm{M}})}{M}}\Big\},$$
$$\mathcal{E}_{\mathrm{L}} := \Big\{\|\mathbf{y}_{\tau,L}(\mathbf{q}) - \mathbf{y}_\tau(\mathbf{q})\|_2 \le C\,\|\mathbf{V}\|_2\sqrt{\tfrac{\log(4/\delta_{\mathrm{L}})}{L}}\Big\}.$$

By the Lemma 7, $\mathbb{P}(\mathcal{E}_{\mathrm{M}}) \ge 1 - \delta_{\mathrm{M}}$, and by Lemma 6, $\mathbb{P}(\mathcal{E}_{\mathrm{L}}) \ge 1 - \delta_{\mathrm{L}}$.

Moreover, Lemma 8 gives the deterministic bound

$$\|\mathbf{y}_\tau(\mathbf{q}) - \mathbf{y}^*(\mathbf{q})\|_2 \le 2B\Big(\frac{1}{Z_{\tau,\min}} + \frac{\sqrt{B}}{Z_{\min}Z_{\tau,\min}}\Big)\varepsilon_\tau(\mathbf{q})\,\|\mathbf{V}\|_2.$$

On the intersection $\mathcal{E}_{\mathrm{M}} \cap \mathcal{E}_{\mathrm{L}}$, substituting these three bounds into equation 45 yields the claimed inequality. Finally, by the union bound,

$$\mathbb{P}(\mathcal{E}_{\mathrm{M}} \cap \mathcal{E}_{\mathrm{L}}) \ge 1 - \delta_{\mathrm{M}} - \delta_{\mathrm{L}}. \quad (46)$$

We conclude the proof by taking $\delta_M = \delta_L = \frac{\delta}{2}$ in eq. equation 46. ■

C Additional Proofs

Proof [Proof of Lemma 4] We first note that $X = \mathbf{q}^\top\mathbf{k} \sim \mathcal{N}(0, \|\mathbf{q}\|_2^2) = \mathcal{N}(0,1)$, hence $\mathbb{E}[X] = 0$ and $\mathbb{E}[X^2] = 1$. Also, for each i, $\hat{\mathbf{w}}_i^\top\mathbf{k} \sim \mathcal{N}(0,1)$ is symmetric about 0, so $\mathbb{E}[\mathrm{sign}(\hat{\mathbf{w}}_i^\top\mathbf{k})] = 0$ and therefore $\mathbb{E}[Y] = \sum_i s_i\,\mathbb{E}[\mathrm{sign}(\hat{\mathbf{w}}_i^\top\mathbf{k})] = 0$. For $\mathbb{E}[Y^2]$, expand $Y^2 = \sum_{i=1}^P s_i^2 + \sum_{i\ne j} s_i s_j\ \mathrm{sign}(\hat{\mathbf{w}}_i^\top\mathbf{k})\,\mathrm{sign}(\hat{\mathbf{w}}_j^\top\mathbf{k})$, so $\mathbb{E}[Y^2] = \sum_{i=1}^P s_i^2 + \sum_{i\ne j} s_i s_j\,\mathbb{E}[\mathrm{sign}(\hat{\mathbf{w}}_i^\top\mathbf{k})\,\mathrm{sign}(\hat{\mathbf{w}}_j^\top\mathbf{k})]$. For $i \ne j$, the pair $(\hat{\mathbf{w}}_i^\top\mathbf{k}, \hat{\mathbf{w}}_j^\top\mathbf{k})$ is jointly Gaussian with correlation $\rho_{ij} = \hat{\mathbf{w}}_i^\top\hat{\mathbf{w}}_j$. Using the standard identity $\mathbb{E}[\mathrm{sign}(U)\,\mathrm{sign}(V)] = \frac{2}{\pi}\arcsin(\rho_{ij})$ for jointly Gaussian (U,V), and the orthogonality assumption $\rho_{ij} = 0$, we get $\mathbb{E}[\mathrm{sign}(\hat{\mathbf{w}}_i^\top\mathbf{k})\,\mathrm{sign}(\hat{\mathbf{w}}_j^\top\mathbf{k})] = 0$; thus all cross terms vanish and $\mathbb{E}[Y^2] = \sum_{i=1}^P s_i^2$.

For $\mathbb{E}[XY]$, by linearity, $\mathbb{E}[XY] = \sum_{i=1}^P s_i\,\mathbb{E}[(\mathbf{q}^\top\mathbf{k})\,\mathrm{sign}(\hat{\mathbf{w}}_i^\top\mathbf{k})]$. Fix i and decompose $\mathbf{k} = r\hat{\mathbf{w}}_i + \mathbf{u}$ where $r := \hat{\mathbf{w}}_i^\top\mathbf{k} \sim \mathcal{N}(0,1)$ and $\mathbf{u} \perp \hat{\mathbf{w}}_i$ is independent of r. Then $\mathrm{sign}(\hat{\mathbf{w}}_i^\top\mathbf{k}) = \mathrm{sign}(r)$ and $\mathbf{q}^\top\mathbf{k} = r(\mathbf{q}^\top\hat{\mathbf{w}}_i) + \mathbf{q}^\top\mathbf{u}$. By independence and symmetry, $\mathbb{E}[\mathbf{q}^\top\mathbf{u}\cdot\mathrm{sign}(r)] = \mathbb{E}[\mathbf{q}^\top\mathbf{u}]\mathbb{E}[\mathrm{sign}(r)] = 0$. Also $r\,\mathrm{sign}(r) = |r|$, so $\mathbb{E}[(\mathbf{q}^\top\mathbf{k})\,\mathrm{sign}(\hat{\mathbf{w}}_i^\top\mathbf{k})] = (\mathbf{q}^\top\hat{\mathbf{w}}_i)\mathbb{E}|r|$. Summing over i yields $\mathbb{E}[XY] = C\sum_{i=1}^P(\mathbf{q}^\top\hat{\mathbf{w}}_i)s_i$ with $C = \mathbb{E}|r| = \sqrt{2/\pi}$.

Finally, since $\mathbb{E}[X] = \mathbb{E}[Y] = 0$, the correlation is $\Gamma = \frac{\mathbb{E}[XY]}{\sqrt{\mathbb{E}[X^2]\mathbb{E}[Y^2]}} = C\,\frac{\sum_i(\mathbf{q}^\top\hat{\mathbf{w}}_i)s_i}{\sqrt{\sum_i s_i^2}}$. Writing $\sum_i(\mathbf{q}^\top\hat{\mathbf{w}}_i)s_i = \mathbf{q}^\top\mathbf{W}^\mathsf{T}\mathbf{s}$ and $\|\mathbf{s}\|_2 = \sqrt{\sum_i s_i^2}$ gives $\Gamma = C\,\mathbf{q}^\top\mathbf{W}^\mathsf{T}\hat{\mathbf{s}}$. ■

Hard vs. soft scoring. Lemma 4 shows that, under approximately orthogonal random planes, the correlation between the true similarity $X = \mathbf{q}^\top\mathbf{k}$ and the aggregated score Y takes the form

$$\Gamma = C\,\mathbf{q}^\top\mathbf{W}^\top\hat{\mathbf{s}} = C\,\frac{(\mathbf{W}\mathbf{q})^\top\mathbf{s}}{\|\mathbf{s}\|_2}, \qquad C = \sqrt{\frac{2}{\pi}},$$

where $\mathbf{s} = (s_1, \ldots, s_P)^\top$ collects the per-plane scores and $\hat{\mathbf{s}} = \mathbf{s}/\|\mathbf{s}\|_2$. Thus, the behavior of the scoring rule is governed by the alignment between the projected query $\mathbf{Wq}$ and the score vector $\mathbf{s}$.

Hard scoring. For hard LSH, $s_i^{\text{hard}} = \text{sign}(\mathbf{q}^\top \hat{\mathbf{w}}_i)$, equivalently $\mathbf{s}^{\text{hard}} = \text{sign}(\mathbf{Wq})$. In this case $\|\mathbf{s}^{\text{hard}}\|_2 = \sqrt{P}$ and

$$\Gamma_{\text{hard}} = C\,\frac{(\mathbf{Wq})^\top \text{sign}(\mathbf{Wq})}{\sqrt{P}} = \frac{C}{\sqrt{P}}\|\mathbf{Wq}\|_1.$$

Hence, hard scoring depends only on the ℓ_1 magnitude of the projected coordinates and discards all directional information within each orthant. As a result, small perturbations in $\mathbf{Wq}$ can induce discontinuous changes in $\hat{\mathbf{s}}^{\text{hard}}$, leading to slow concentration and unstable rankings at finite L.

Soft scoring. For soft scoring, $s_i^{\text{soft}} = \tanh(\mathbf{q}^\top \hat{\mathbf{w}}_i)$, i.e., $\mathbf{s}^{\text{soft}} = \tanh(\mathbf{Wq})$ (applied elementwise). The corresponding correlation is

$$\Gamma_{\text{soft}} = C\,\frac{(\mathbf{Wq})^\top \tanh(\mathbf{Wq})}{\|\tanh(\mathbf{Wq})\|_2}.$$

In the small-signal regime typical of high-dimensional random projections, $|(\mathbf{Wq})_i| \ll 1$ and $\tanh(\mathbf{Wq}) \approx \mathbf{Wq}$, yielding

$$\Gamma_{\text{soft}} \;\approx\; C\,\|\mathbf{Wq}\|_2, \qquad \hat{\mathbf{s}}^{\text{soft}} \;\approx\; \frac{\mathbf{Wq}}{\|\mathbf{Wq}\|_2}.$$

Thus, soft scoring preserves the directional structure of the projected coordinates and varies smoothly with $\mathbf{q}$.

Comparison. By the inequality $\|\mathbf{x}\|_1 \le \sqrt{P}\|\mathbf{x}\|_2$, we have

$$\Gamma_{\text{hard}} = \frac{C}{\sqrt{P}}\|\mathbf{Wq}\|_1 \;\le\; C\,\|\mathbf{Wq}\|_2 \;\approx\; \Gamma_{\text{soft}},$$

with strict inequality unless the coordinates of $\mathbf{Wq}$ have equal magnitude. Consequently, soft scoring achieves strictly stronger alignment with the true similarity signal in the regime relevant for ranking-based inference, explaining its faster concentration and improved top-k stability at finite L.

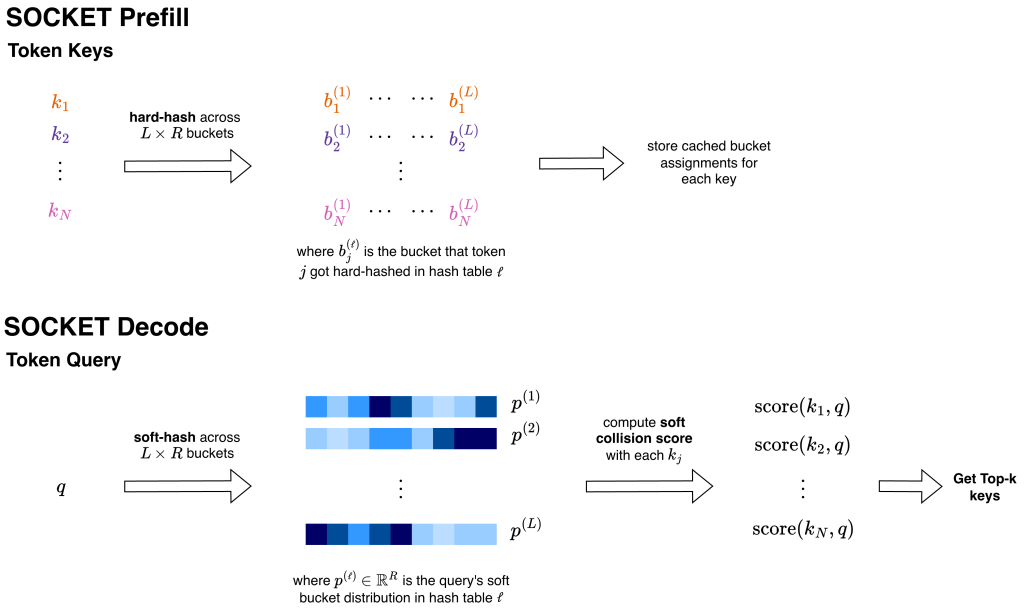


Figure 4: Overview of the SOCKET design and workflow.