SUBCENSA: TRAINING-FREE BLOCK-SPARSE DECODING WITH COUNT-WEIGHTED SUB-CENTROID MASS ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Block-sparse attention accelerates long-context decoding by reading only selected contiguous key-value (KV) blocks, while its quality and efficiency depend on identifying the blocks carrying most of the dense attention mass without scanning their original keys. We introduce **Sub-Centroid Sparse Attention (SubCenSA)**, a training-free method that represents each key block with a few count-weighted sub-centroids, retaining relevant within-block structure at low storage and scoring cost. SubCenSA retains the full KV cache and block-aligned I/O, and computes standard softmax attention over the original keys and values in the selected read set. In exact arithmetic, each per-head estimate lower-bounds the true block mass and is at least as tight as mean-centroid pooling. For the shared grouped-query attention (GQA) selector, we show that the cross-atom oscillation of these residuals controls the self-normalized group-score error, yielding margin-based guarantees for recovery of the true group top-m blocks, retained group mass, and weighted fixed-state output error. We evaluate SubCenSA through selector-fidelity diagnostics, downstream performance on RULER, LongBench, and LongBench-V2-CoT, design ablations, and operator-level and end-to-end profiling at contexts up to 256K. SubCenSA improves over classical block indices by up to 9.08 RULER points and achieves 50.95 on LongBench and 40.95 on LongBench-V2-CoT. At a 6.25% nominal complete-block KV budget, it remains at most 99.4% of dense-attention quality while delivering up to $5.2\times$ operator and **[TBD]**$\times$ end-to-end decoding speedups over dense FlashAttention-2.

1 INTRODUCTION

Large language models are increasingly required to reason over long documents (Pang et al., 2022), code repositories (Zhang et al., 2023a), and retrieval-augmented corpora (Lewis et al., 2020). During autoregressive generation, dense softmax attention reads the keys and values of all preceding tokens for each new token, causing key-value (KV) cache traffic to grow linearly with context length. IO-aware exact kernels such as FlashAttention (Dao et al., 2022) and FlashAttention-2 (Dao, 2024) reduce intermediate memory operations but still access the complete cache, making repeated KV reads a primary bottleneck at long contexts.

Query-aware sparse attention (Tang et al., 2024; Singhania et al., 2024; Liu et al., 2024a;b) reduces this traffic by reading only a small fraction of the cache for the current query. We study a training-free block-sparse adaptation for pretrained LLMs that retains the full cache, partitions it into contiguous blocks, and reads only a small selected set of blocks without updating model parameters or evicting cached entries. Contiguous block reads facilitate coalesced memory access and avoid token-granular gathers, while a compact block-level index avoids scoring every cached token. The central challenge is therefore to identify a small set of blocks whose retrieval closely preserves the dense-attention computation.

A principled value-agnostic proxy for this objective is block attention mass—the total unnormalized attention weight assigned to a block under the current query (Tzachristas et al., 2025; Tian et al., 2026; Hwang, 2026). Computing it exactly, however, still requires scoring every key and would erode the benefit of sparse decoding. Existing selectors instead estimate block mass from compact

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.

pooled, bounding, second-order, or spectral statistics (Tang et al., 2024; Lu et al., 2025; Tian et al., 2026; Wang et al., 2026a;b; Hwang, 2026). These representations offer different trade-offs among metadata, query-time cost, and estimation fidelity, particularly for blocks with heterogeneous keys.

Within this design space, we introduce **Sub-Centroid Sparse Attention (SubCenSA)**, which summarizes each block with a few count-weighted sub-centroids. Each centroid represents a group of keys, and its count preserves the group's contribution when estimating block attention mass. This representation captures heterogeneous within-block structure at modest storage and scoring cost. The subclusters serve only as selection metadata: the original KV cache is retained in its existing layout, and standard softmax attention is computed over the original entries in the selected contiguous blocks.

The count-weighted construction also provides a direct connection between mass estimation and selection quality. For each query head, the SubCenSA estimate lower-bounds the true block mass and is no smaller than the mean-centroid estimate. Its one-sided residual yields margin-based guarantees for single-head block ordering, top-m recovery, and retained mass. The deployed GQA selector self-normalizes these per-head estimates and selects one block set shared by the query heads associated with each KV head. Because normalized probabilities are invariant to an offset shared by all normalization atoms of a head, the relevant group-level error is controlled by the cross-atom oscillation of the per-head residuals, with the exact mandatory atom providing the zero-residual anchor. We use this observation to derive corresponding guarantees for recovery of the true group top-m set, retained group mass, and the weighted fixed-state output error of the shared-set selector.

We evaluate SubCenSA along four complementary dimensions: (i) selector-ranking and mass-retention diagnostics, (ii) downstream performance on RULER (Hsieh et al., 2024), LongBench (Bai et al., 2024), and LongBench-V2-CoT (Bai et al., 2025), (iii) controlled design ablations, and (iv) operator and end-to-end efficiency at contexts up to 256K. This progression tests whether improved mass estimation carries through to long-context generation quality. SubCenSA improves over classical block indices by up to 9.08 RULER points and achieves 50.95 on LongBench and 40.95 on LongBench-V2-CoT. At a 6.25% nominal complete-block KV budget, it remains within 6.16 points of dense-attention quality while delivering up to $5.2\times$ attention-operator and **[TBD]**$\times$ end-to-end decoding speedups over matched dense FlashAttention-2.

Our contributions are: (1) **A training-free sub-centroid mechanism for block-sparse decoding.** SubCenSA estimates block attention mass using a few count-weighted sub-centroids while retaining the full KV cache, contiguous block reads, and exact selected-block attention. (2) **A ranking-oriented theoretical analysis spanning per-head estimation and shared GQA selection.** We connect one-sided mass residuals to block ranking and retained mass, and show how their cross-block oscillation controls the self-normalized group selector used in GQA. (3) **An empirical evaluation connecting selector fidelity, downstream quality, and decoding efficiency.** Experiments cover diverse long-context benchmarks, controlled design ablations, and operator-level and end-to-end profiling at contexts up to 256K.

2 Background and Related Work

Unified attention formulation. Consider one query head with query $\mathbf{q} \in \mathbb{R}^{d_h}$ and cached key–value pairs $\{(\mathbf{k}_i, \mathbf{v}_i)\}_{i=1}^{T}$. For block size b, let $\mathcal{B} = \{B_1, \dots, B_N\}$ be the $N = \lfloor T/b \rfloor$ complete contiguous blocks and let P_{tail} denote the final $r = T - Nb$ tokens when $r > 0$. The cache atoms are $\mathcal{X}_{\text{cache}} = \mathcal{B} \cup \{P_{\text{tail}}\}$ when the tail is non-empty, and $\mathcal{X}_{\text{cache}} = \mathcal{B}$ otherwise. The query–key logit is $s_i = \mathbf{q}^\top \mathbf{k}_i / \sqrt{d_h}$. For any non-empty set of read atoms $\mathcal{R}(\mathbf{q}) \subseteq \mathcal{X}_{\text{cache}}$, the attention output is

$$\mathbf{o}_{\mathcal{R}} = \frac{1}{Z_{\mathcal{R}}} \sum_{X \in \mathcal{R}(\mathbf{q})} \sum_{i \in X} \exp(s_i)\mathbf{v}_i, \qquad Z_{\mathcal{R}} = \sum_{X \in \mathcal{R}(\mathbf{q})} \sum_{i \in X} \exp(s_i). \tag{1}$$

Dense attention corresponds to $\mathcal{R}(\mathbf{q}) = \mathcal{X}_{\text{cache}}$; its output and normalizer are denoted by $\mathbf{o}$ and Z. Block-sparse attention (Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020) instead reads a strict subset of the complete blocks while always reading the non-empty partial tail directly. It applies the same softmax attention to the original keys and values in the resulting read atoms. As the context grows, this formulation reduces KV traffic in proportion to the number of complete blocks read, while retaining contiguous block-level memory access (Dao et al., 2022; Ohayon et al., 2025).

Block selection and attention mass. A lightweight index assigns each complete block a query-dependent score $g(\mathbf{q}, B)$. Let $\mathcal{A}_{\text{full}}(\mathbf{q}) \subseteq \mathcal{B}$ contain the mandatory complete blocks, such as sink or recent blocks (Xiao et al., 2024), and let $\mathcal{O} = \mathcal{B} \setminus \mathcal{A}_{\text{full}}$ be the optional complete blocks. With optional budget m, the selector forms

$$\mathcal{S}_m(\mathbf{q}) = \text{TopKBlocks}_m \{(B, g(\mathbf{q}, B)) : B \in \mathcal{O}\}, \qquad \mathcal{R}(\mathbf{q}) = \mathcal{A}(\mathbf{q}) \cup \mathcal{S}_m(\mathbf{q}), \tag{2}$$

where $\mathcal{A} = \mathcal{A}_{\text{full}} \cup \{P_{\text{tail}}\}$ when the tail is non-empty and $\mathcal{A} = \mathcal{A}_{\text{full}}$ otherwise. Thus the tail is an exact mandatory atom but never an indexed top-m candidate. A principled value-agnostic selection target is the block attention mass

$$M(\mathbf{q}, B) = \sum_{i \in B} e^{s_i}, \qquad L(\mathbf{q}, B) = \log M(\mathbf{q}, B). \tag{3}$$

Because the dense normalizer Z is common to all blocks, $M(\mathbf{q}, B)$, $L(\mathbf{q}, B)$, and the normalized block probability $M(\mathbf{q}, B)/Z$ induce the same ranking. Let $\alpha(\mathcal{R}) = Z_{\mathcal{R}}/Z$ denote the fraction of dense attention mass retained by a read set. At a fixed attention state, for a non-empty proper read set define $\overline{\mathcal{R}} = \mathcal{X}_{\text{cache}} \setminus \mathcal{R}$ and let $\mathbf{o}_{\overline{\mathcal{R}}}$ be its normalized attention output. Then

$$\mathbf{o} - \mathbf{o}_{\mathcal{R}} = \big(1 - \alpha(\mathcal{R})\big)\big(\mathbf{o}_{\overline{\mathcal{R}}} - \mathbf{o}_{\mathcal{R}}\big), \qquad \|\mathbf{o} - \mathbf{o}_{\mathcal{R}}\|_2 \le 2V\big(1 - \alpha(\mathcal{R})\big) \tag{4}$$

whenever $\|\mathbf{v}_i\|_2 \le V$ for all tokens (Tzachristas et al., 2025; Tian et al., 2026; Hwang, 2026). For the dense read set the error is zero directly, without invoking the complement output. At a fixed attention state, with a fixed mandatory atom set, equally sized complete blocks, equal per-block read cost, and fixed optional cardinality m, selecting the optional blocks with the largest true masses maximizes retained mass and minimizes the corresponding value-agnostic worst-case bound. Block mass therefore provides a well-defined target under these conditions.

Shared selection under grouped-query attention. Let G contain the query heads sharing one KV head. For any cache atom or disjoint union of atoms X, let $p_h(X) = M_h(X)/Z_h$ be its normalized mass for head h. For a common read set, we use the group objective

$$J_G(X) = \sum_{h \in G} \beta_h p_h(X), \qquad J_G(\mathcal{R}) = \sum_{X \in \mathcal{R}} J_G(X), \qquad \beta_h > 0, \quad \sum_{h \in G} \beta_h = 1, \tag{5}$$

with $\beta_h = 1/|G|$ by default. Heads assigned zero weight may equivalently be omitted from G. Applying Eq. equation 4 to each head yields

$$\sum_{h \in G} \beta_h \|\mathbf{o}_h - \mathbf{o}_{h,\mathcal{R}}\|_2 \le 2V\big(1 - J_G(\mathcal{R})\big). \tag{6}$$

Consequently, ranking optional blocks by $J_G(B)$ maximizes the retained group mass for a shared block set. This bound concerns the weighted per-head outputs before output projection. Section 3 introduces SubCenSA's self-normalized estimate of J_G and analyzes how its residuals affect the resulting shared-set ranking.

Relation to prior work. Cache-eviction methods retain selected entries (Zhang et al., 2023b; Li et al., 2024); retrieval methods fetch individual tokens or semantic groups (Singhania et al., 2024; Liu et al., 2024a;b); and learned methods train sparse routers (Yuan et al., 2025; Lu et al., 2025; Zhao et al., 2025). Compact block selectors instead use pooled representatives (Lu et al., 2025), coordinate bounds (Tang et al., 2024), second-order statistics (Wang et al., 2026a; Tian et al., 2026), or spectral summaries (Wang et al., 2026b; Hwang, 2026). SubCenSA belongs to this last family but retains the full cache and contiguous block layout, preserves within-block heterogeneity with count-weighted sub-centroids, and requires no learned router or model update.

3 Method: Sub-Centroid Sparse Attention

SubCenSA modifies the selection stage of block-sparse decoding. As illustrated in Figure 1, each completed block is represented by a few count-weighted sub-centroids, which estimate per-head block masses and support self-normalized shared selection under GQA. The KV cache and contiguous block layout are retained, and standard softmax attention is computed over the original keys and values in the selected read atoms. We call this computation *exact selected-block attention*: exactness is relative to the selected original KV entries, not to dense attention, bitwise equivalence, or the subsequent autoregressive trajectory.

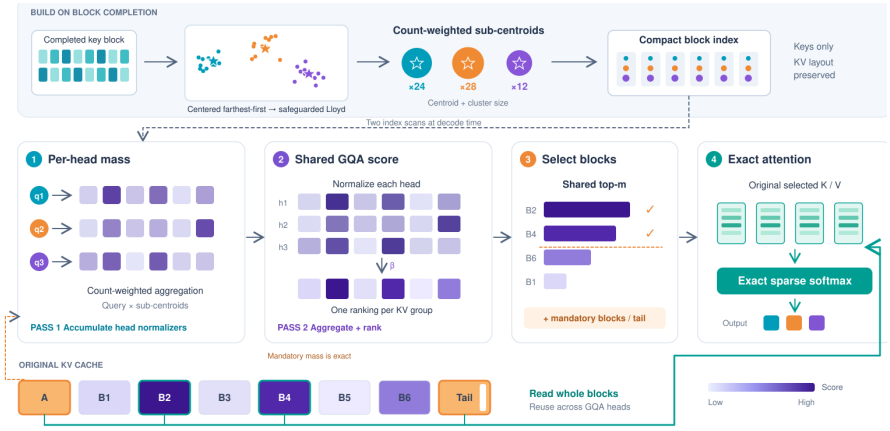


Figure 1: Overview of the SubCenSA architecture. The KV cache is partitioned into fixed-size blocks, and each completed block is summarized by K_c sub-centroids with counts $(\mathbf{c}_{B,j}, n_{B,j})$, built by centered farthest-first seeding followed by two safeguarded Lloyd updates. At decode time, blocks are scored with the count-weighted log-sum-exp $\widehat{L}_{\mathrm{SC}}$ of Eq. equation 8 and ranked by the self-normalized group score of Eq. equation 13 shared within GQA groups; exact selected-block attention is computed over the original KV entries in the resulting read set.

3.1 Count-Weighted Sub-Centroid Mass Estimation

Write $\mathbf{a} = \mathbf{q}/\sqrt{d_h}$. For a block B with b contiguous keys, SubCenSA constructs a partition $\mathcal{P}_B = \{\mathcal{C}_{B,1}, \ldots, \mathcal{C}_{B,K_c}\}$ into K_c non-empty subclusters, then stores their counts and centroid vectors:

$$n_{B,j} = |\mathcal{C}_{B,j}|, \qquad \mathbf{c}_{B,j} = \frac{1}{n_{B,j}} \sum_{i \in \mathcal{C}_{B,j}} \mathbf{k}_i, \qquad \mathcal{I}_B = \{(\mathbf{c}_{B,j}, n_{B,j})\}_{j=1}^{K_c}. \tag{7}$$

The corresponding mass and log-mass estimates are

$$\widehat{M}_{\mathrm{SC}}(\mathbf{q}, B) = \sum_{j=1}^{K_c} n_{B,j} e^{\mathbf{a}^\top \mathbf{c}_{B,j}}, \qquad \widehat{L}_{\mathrm{SC}}(\mathbf{q}, B) = \mathrm{LSE}_j \left(\mathbf{a}^\top \mathbf{c}_{B,j} + \log n_{B,j} \right). \tag{8}$$

The count makes each centroid represent the aggregate contribution of its assigned keys rather than an equally weighted prototype. The subclusters are used only for scoring; the enclosing temporal block remains the physical selection and I/O unit. When $K_c = 1$, Eq. equation 8 reduces to the mean-centroid estimate $\widehat{M}_{\mathrm{mean}}(\mathbf{q}, B) = b e^{\mathbf{a}^\top \boldsymbol{\mu}_B}$, where $\boldsymbol{\mu}_B = b^{-1} \sum_{i \in B} \mathbf{k}_i$. When $K_c = b$, every key forms a singleton and the estimate equals the exact block mass. Without count weighting, neither the improvement over mean pooling nor the mass ordering is preserved in general; Appendix F provides explicit counterexamples. For the temperature ablation, we use

$$\widehat{L}^{(\tau)}(\mathbf{q}, B) = \tau \log \sum_{j=1}^{K_c} n_{B,j} \exp\left(\frac{\mathbf{a}^\top \mathbf{c}_{B,j}}{\tau} \right), \qquad \tau > 0.$$

Only the default $\tau = 1$ is the count-weighted empirical-mass estimator in Eq. equation 8; other temperatures are alternative ranking scores rather than estimates of that same mass.

Define the log-mass residual $e_B = L(\mathbf{q}, B) - \widehat{L}_{\mathrm{SC}}(\mathbf{q}, B)$. For each subcluster, let $r_{B,j} = \max_{i \in \mathcal{C}_{B,j}} |\mathbf{a}^\top (\mathbf{k}_i - \mathbf{c}_{B,j})|$, $r_B = \max_j r_{B,j}$, $D_B = \max_j \max_{i \in \mathcal{C}_{B,j}} \|\mathbf{k}_i - \mathbf{c}_{B,j}\|_2$, and $\pi_{B,j} = n_{B,j} e^{\mathbf{a}^\top \mathbf{c}_{B,j}} / \widehat{M}_{\mathrm{SC}}(\mathbf{q}, B)$.

Theorem 1 (Estimator hierarchy and residual control)**.** *In exact arithmetic, for any query, block, and non-empty partition whose centroids are the exact empirical means of their subclusters,*

$$\widehat{M}_{\mathrm{mean}}(\mathbf{q}, B) \le \widehat{M}_{\mathrm{SC}}(\mathbf{q}, B) \le M(\mathbf{q}, B). \tag{9}$$

If $\mathcal{P}'$ refines $\mathcal{P}$, then $\widehat{M}_{\mathcal{P}} \le \widehat{M}_{\mathcal{P}'} \le M$. Moreover,

$$0 \le e_B \le \log \sum_j \pi_{B,j} e^{r_{B,j}^2/2} \le \frac{r_B^2}{2} \le \frac{\|\mathbf{a}\|_2^2 D_B^2}{2}. \tag{10}$$

Thus, SubCenSA retains variation represented between subclusters, while its remaining error is controlled by the variation within them. Refinement monotonicity applies to nested partitions; independently constructed solutions for different K_c need not be nested. Appendix E proves the hierarchy and derives construction-dependent certificates that connect the residual to selection quality.

3.2 Ranking Guarantees and Shared GQA Selection

Single-head selection. At a fixed attention state, let $\mathcal{O} = \mathcal{B} \setminus \mathcal{A}_{\text{full}}$ be the optional complete blocks, $\mathcal{S}_m^\star$ their true top-m set under $L_B = L(\mathbf{q}, B)$, and $\widehat{\mathcal{S}}_m$ the set selected under $\widehat{L}_{\text{SC}}(\mathbf{q}, B)$. Assume equally sized complete blocks, equal per-block read cost, a fixed mandatory atom set, exact arithmetic, a common deterministic tie-breaking rule, and $1 \le m < |\mathcal{O}|$; the cases $m = 0$ and $m = |\mathcal{O}|$ are trivial. Define

$$\Delta_m = \min_{\substack{B \in \mathcal{S}_m^\star \\ C \in \mathcal{O} \setminus \mathcal{S}_m^\star}} (L_B - L_C), \qquad \epsilon^\star = \max_{B \in \mathcal{S}_m^\star} e_B, \tag{11}$$

with $\Delta_m = 0$ when the boundary is tied. For a block set S, write $M(S) = \sum_{B \in S} M(\mathbf{q}, B)$.

Theorem 2 (Single-head ranking and retained mass)**.** *Under the fixed-state, equal-cost, exact-arithmetic, and tie-breaking conditions above, if $L_B > L_C$ and $e_B < L_B - L_C$, then $\widehat{L}_{\text{SC}}(\mathbf{q}, B) > \widehat{L}_{\text{SC}}(\mathbf{q}, C)$. Consequently,*

$$\epsilon^\star < \Delta_m \quad \Longrightarrow \quad \widehat{\mathcal{S}}_m = \mathcal{S}_m^\star, \qquad M(\widehat{\mathcal{S}}_m) \ge e^{-\epsilon^\star} M(\mathcal{S}_m^\star). \tag{12}$$

The theorem converts the blockwise residual into two selector-level statements: exact recovery under a margin condition and a retained-mass guarantee otherwise. Combined with Eq. equation 4, the latter yields the corresponding fixed-state attention-output bound. Appendix E.5 gives the proof and develops additive regret, rank-correlation, and offline certification results.

Self-normalized GQA selection. Let G denote the query heads sharing one KV head. SubCenSA treats the mandatory read, including any partial tail, as an exact atom and estimates

$$\widehat{Z}_h = M_h(\mathcal{A}) + \sum_{B \in \mathcal{O}} \widehat{M}_{\text{SC},h}(B), \qquad \widehat{p}_h(B) = \frac{\widehat{M}_{\text{SC},h}(B)}{\widehat{Z}_h}, \qquad \widehat{J}_G(B) = \sum_{h \in G} \beta_h \widehat{p}_h(B). \tag{13}$$

where $\beta_h > 0$, $\sum_{h \in G} \beta_h = 1$, and $\beta_h = 1/|G|$ by default; a zero-weight head is omitted from G. The selector ranks blocks by $g_G(B) = \log \widehat{J}_G(B)$, computed as

$$g_G(B) = \text{LSE}_{h \in G}\big[\widehat{L}_{\text{SC},h}(B) - \log \widehat{Z}_h + \log \beta_h\big].$$

This estimates the group objective J_G in Eq. equation 5. Normalized probabilities are invariant to a head-specific additive shift applied uniformly to the log masses of every normalization atom for that head.

For the selection atoms $\mathcal{X} = \{\mathcal{A}\} \cup \mathcal{O}$, where $\mathcal{A}$ is the exact aggregate of all mandatory reads, define

$$e_{h,x} = \log M_h(x) - \log \widehat{M}_h(x), \qquad \omega_h = \max_{x \in \mathcal{X}} e_{h,x} - \min_{x \in \mathcal{X}} e_{h,x}, \qquad \Omega = \max_{h \in G} \omega_h, \tag{14}$$

where $\widehat{M}_h(\mathcal{A}) = M_h(\mathcal{A})$. Let $\mathcal{S}_{G,m}^\star$ and $\widehat{\mathcal{S}}_{G,m}$ be the true and estimated top-m optional sets under J_G and $\widehat{J}_G$, and define $\Delta_{G,m} = \min_{\substack{B \in \mathcal{S}_{G,m}^\star \\ C \in \mathcal{O} \setminus \mathcal{S}_{G,m}^\star}} \big[\log J_G(B) - \log J_G(C)\big]$. We again assume a fixed attention state, equally sized complete blocks, equal per-block read cost, exact arithmetic, common deterministic tie-breaking, and $1 \le m < |\mathcal{O}|$.

Theorem 3 (Self-normalized GQA ranking)**.** *Under these conditions, for every optional block B,*

$$e^{-\Omega} J_G(B) \le \widehat{J}_G(B) \le e^{\Omega} J_G(B). \tag{15}$$

Therefore,

$$\Delta_{G,m} > 2\Omega \implies \widehat{\mathcal{S}}_{G,m} = \mathcal{S}_{G,m}^\star, \qquad J_G(\widehat{\mathcal{S}}_{G,m}) \ge e^{-2\Omega} J_G(\mathcal{S}_{G,m}^\star). \tag{16}$$

Under the common value-norm bound V,

$$\sum_{h \in G} \beta_h \left\| \mathbf{o}_h - \mathbf{o}_{h,\mathcal{A} \cup \widehat{\mathcal{S}}_{G,m}} \right\|_2 \le 2V \left[1 - J_G(\mathcal{A}) - e^{-2\Omega} J_G(\mathcal{S}_{G,m}^\star)\right]. \tag{17}$$

Table 1: Persistent index, per-head score, and logical storage per complete block and KV head. The score is shown before the common self-normalized GQA aggregation. Byte formulas exclude temporary buffers, allocator padding, and shared layer-level projections.

Method	Stored index	Per-head log score $\ell_h(B)$	Logical persistent bytes
ExactBlockMass	none; reads the original block keys	$\mathrm{LSE}_{i\in B}[\mathbf{a}_h^\top \mathbf{k}_i]$	0 auxiliary; $bd_h s_k$ key bytes scanned
Centroid-Index	mean $\boldsymbol{\mu}_B$	$\log b + \mathbf{a}_h^\top \boldsymbol{\mu}_B$	$d_h s_c$
Quest-Index	coordinate envelope $(\mathbf{k}_B^{\min}, \mathbf{k}_B^{\max})$	$\log b + \sum_\ell \max\{a_{h\ell} k_{B\ell}^{\min}, a_{h\ell} k_{B\ell}^{\max}\}$	$2d_h s_c$
SPLA-Index	mean $\boldsymbol{\mu}_B$ and diagonal variance $\mathbf{v}_B$	$\log b + \mathbf{a}_h^\top \boldsymbol{\mu}_B + \log(1 + \frac{1}{2}\sum_\ell a_{h\ell}^2 v_{B\ell})$	$2d_h s_c$
COBS-Index (r)	mean and r compressed covariance factors defining $\widehat{\boldsymbol{\Sigma}}_B$	$\log b + \mathbf{a}_h^\top \boldsymbol{\mu}_B + \frac{1}{2}\mathbf{a}_h^\top \widehat{\boldsymbol{\Sigma}}_B \mathbf{a}_h$	$d_h s_c + r d_f s_f + r s_{\mathrm{sc}}$
LOCKS-Index (r)	mean, local basis $\mathbf{V}_B$, and token coefficients $\{\mathbf{c}_{B,i}\}$	$\mathrm{LSE}_{i\in B}[\mathbf{a}_h^\top \boldsymbol{\mu}_B + (\mathbf{V}_B^\top \mathbf{a}_h)^\top \mathbf{c}_{B,i}]$	$d_h s_\mu + r d_h s_V + b r s_C + S_{\mathrm{sc}}$
SubCenSA (K_c)	K_c empirical centroids and counts $(\mathbf{c}_{B,j}, n_{B,j})$	$\mathrm{LSE}_j[\mathbf{a}_h^\top \mathbf{c}_{B,j} + \log n_{B,j}]$	$K_c(d_h s_c + s_n)$

Self-normalization is insensitive to a residual offset only when that offset is shared by every normalization atom of a head. In SubCenSA the exact mandatory atom has zero residual and therefore anchors this comparison; the resulting bound is controlled by residual oscillation across the mandatory and optional atoms rather than by a free cancellation of optional-block bias. Theorem 3 extends the top-m and retained-mass guarantees to the shared selector used by SubCenSA. Its output bound concerns the weighted per-head errors before output projection. Appendix E.6 proves the result and gives additive, total-variation, and realized-read variants; Appendix G studies alternative cross-head aggregation rules.

3.3 Index Construction and Decode-Time Scoring

Completed blocks use centered farthest-first seeding and deterministic empty-cluster repair, followed by two Lloyd assignment–mean updates. This yields three snapshots in total—the initial snapshot and two updated snapshots—and SubCenSA returns the last one whose covering radius does not exceed the initial radius. The index is built from transformed cached keys in FP32 and then stored in the chosen format. With $K_c = 2$, $b = 64$, FP16 centroids, and 16-bit counts, it adds approximately 1.6% of the FP16 KV bytes. Complete prompt blocks are indexed in one batch, and a new summary is appended only when the tail fills a block.

At decode time, one index scan accumulates $\widehat{Z}_h$ and a second forms $g_G(B)$ while maintaining the top-m candidates. The sparse kernel then performs exact selected-block attention over the original KV entries. For N complete blocks, index storage is $O(N H_{\mathrm{kv}} K_c d_h)$, scoring is $O(N H_q K_c d_h)$ per query token, and selection keeps $O(m)$ state per group. Construction, kernels, and finite-precision effects are detailed in Appendices A, B, and G.3.

4 Experiments

We evaluate selector fidelity, downstream quality, decoding efficiency, and key design choices. Unless stated otherwise, SubCenSA uses $b = 64$, $K_c = 2$, $\tau = 1$, the last radius-feasible snapshot, and the self-normalized GQA score in Eq. equation 13; all choices are fixed on an independent development split. Prefill is dense and sparsity begins at single-token decoding. For Qwen3.5, only the full-softmax layers are replaced.

RULER uses complete-block budgets of 4,096, 8,192, and 16,384 tokens at 64K, 128K, and 256K; LongBench uses 15%, and LongBench-V2-CoT uses 1,024 tokens. Mandatory sink and recent blocks count toward the budget, while an incomplete tail of at most 63 tokens is read additionally. All compact indices share the same budget, normalized GQA aggregation, and selected- block attention backend; "-Index" denotes the corresponding statistic inside this common wrapper. Appendix C gives the complete protocols and baseline definitions.

Table 1 makes the index-cost comparison explicit. Here $\mathbf{a}_h = \mathbf{q}_h/\sqrt{d_h}$; d_f is the stored COBS factor dimension; $s_k, s_c, s_n, s_f, s_\mu, s_V, s_C$ are bytes per corresponding scalar; and s_{sc} and S_{sc} account for per-factor and total quantization scales. With $b = 64$, $d_h = 128$, FP16 vector statistics, and uint16 counts, one FP16 KV block occupies 32,768 bytes per KV head, whereas Centroid-, Quest-, SPLA-, and default SubCenSA indices occupy 256 (0.78%), 512 (1.56%), 512 (1.56%), and 516 bytes (1.57%), respectively. ExactBlockMass stores no auxiliary index but scores all b original keys at $O(bd_h)$ cost; the four compact scores cost $O(d_h)$, $O(d_h)$, $O(d_h)$, and $O(K_c d_h)$. COBS and LOCKS additionally read low-rank factors (and, for LOCKS, token coefficients), giving $O(d_h + r d_f)$ and

Table 2: Selector fidelity of the default $K_c = 2$ configuration on Qwen3.5-9B RULER-128K. Results are aggregated over realized sparse decoding trajectories.

Method	MassRatio total	MassRatio optional	Recall@m	Mass-NDCG
Centroid-Index	**[TBD]**	**[TBD]**	**[TBD]**	**[TBD]**
Quest-Index	**[TBD]**	**[TBD]**	**[TBD]**	**[TBD]**
SPLA-Index	**[TBD]**	**[TBD]**	**[TBD]**	**[TBD]**
SubCenSA ($K_c = 2$)	**[TBD]**	**[TBD]**	**[TBD]**	**[TBD]**

$O(d_h + r(d_h + b))$ scoring. Because their factor quantization and scales are packing-dependent, equal-storage comparisons use measured packed bytes rather than nominal vector counts.

4.1 Selector Fidelity

We measure recovery of J_G on realized sparse-decoding trajectories. For the exact and estimated top-m optional sets $\mathcal{S}^{\star}_{G,m}$ and $\widehat{\mathcal{S}}_{G,m}$, the primary metric is

$$\mathrm{MassRatio}_{\mathrm{total}} = \frac{J_G(\mathcal{A}) + J_G(\widehat{\mathcal{S}}_{G,m})}{J_G(\mathcal{A}) + J_G(\mathcal{S}^{\star}_{G,m})}. \tag{18}$$

We also report optional-set mass ratio, Recall@m, and mass-NDCG. Retained mass is primary because swapping nearly tied blocks can lower recall without materially changing the objective. Layer-wise results appear in Appendix C.3.

SubCenSA reaches a total retained-mass ratio of **[TBD]**, improving over the strongest compared index by **[TBD]**; its optional-mass ratio, Recall@m, and mass-NDCG are **[TBD]**, **[TBD]**, and **[TBD]**. These results connect mass estimation to downstream evaluation.

4.2 Downstream Quality

RULER evaluates retrieval and structured reasoning up to 256K; LongBench and LongBench-V2-CoT cover long-document tasks and longer reasoning trajectories. *ExactBlockMass* scores every key to obtain current-state block masses, so it is a fixed-state reference rather than a downstream upper bound when sparse trajectories diverge.

Table 3: Equal-storage RULER-256K comparisons ($b = 64$, 16,384-token budget). Storage is matched within each pair.

Configuration	RULER accuracy (%)
COBS-Index, $r = 4$	**[TBD]**
SubCenSA, $K_c = 4$	76.15
LOCKS-Index, $r = 2$	**[TBD]**
SubCenSA, $K_c = 5$	76.77

Across seven completed RULER settings, SubCenSA is the strongest compact index, gaining 2.00–9.08 points over the best alternative; the largest gains are 9.08 and 8.92 points on Qwen3-8B at 64K and 128K.

On LongBench, SubCenSA scores 50.95: 0.06 points above the strongest compact baseline and 1.03 below dense attention. We treat the 0.06-point difference as descriptive pending paired uncertainty estimates. On LongBench-V2-CoT, its 40.95 score exceeds Quest and SPLA by 0.19 and 0.59 points, but trails Centroid-Index and dense attention by 0.80 and 1.59 points. SubCenSA is therefore competitive on both evaluations, although the larger RULER gains do not uniformly transfer.

Equal-storage comparison. Table 3 raises K_c only to match persistent per-block, per-KV-head storage with COBS $r = 4$ and LOCKS $r = 2$; these variants are excluded from the main quality and profiling results. SubCenSA $K_c = 4$ scores 76.15 versus **[TBD]** for COBS, and $K_c = 5$ scores 76.77 versus **[TBD]** for LOCKS, differences of **[TBD]** and **[TBD]** points. This separates the deployment default from representation quality at equal index capacity.

Table 4: Downstream quality (↑). Panel (a) reports RULER at a 6.25% nominal complete-block KV budget. Panel (b) reports Qwen3-30B-A3B on LongBench at a 15% budget and LongBench-V2-CoT at a 1,024-token budget. SubCenSA uses the default $K_c = 2$ configuration. Bold denotes the best completed compact-index result in each setting.

(a) RULER

	Qwen3-30B-A3B			**Qwen3-8B**		**Qwen3.5-9B**		
Method	64K	128K	256K	64K	128K	64K	128K	256K
KV budget	4,096	8,192	16,384	4,096	8,192	4,096	8,192	16,384
Dense attention	86.15	81.08	79.54	55.54	55.54	85.08	74.77	80.31
ExactBlockMass	86.15	82.15	79.54	56.15	53.85	82.62	73.69	78.31
Centroid-Index	76.77	75.08	66.92	38.92	39.38	74.92	67.69	74.46
Quest-Index	79.85	78.62	72.92	43.23	40.31	73.85	66.31	71.08
SPLA-Index	80.31	78.00	69.69	40.92	40.46	77.54	68.77	75.54
SubCenSA ($K_c = 2$)	**83.38**	**80.62**	**74.92**	**52.31**	**49.38**	**80.77**	**70.77**	-

(b) Qwen3-30B-A3B LongBench evaluations

Benchmark	Dense attention	ExactBlock Mass	Centroid-Index	Quest-Index	SPLA-Index	**SubCenSA** ($K_c = 2$)
LongBench (15%)	51.98	51.51	<u>50.89</u>	50.60	50.87	**50.95**
LongBench-V2-CoT (1,024)	42.54	44.14	**41.75**	40.76	40.36	<u>40.95</u>

Table 5: Matched decoding efficiency. Operator latency is milliseconds per batch step (↓); end-to-end TPOT is milliseconds per generated token (↓). Dense attention uses FlashAttention-2.

Setting	Batch	**Operator**			**End-to-end TPOT**		
		FA2	SubCenSA	Speedup	FA2	SubCenSA	Speedup
Qwen3-30B-A3B, 64K	64	7.588	1.8487	4.104×	**[TBD]**	**[TBD]**	**[TBD]**×
Qwen3-30B-A3B, 128K	64	15.040	2.8906	5.203×	**[TBD]**	**[TBD]**	**[TBD]**×

4.3 Decoding Efficiency

We profile the attention operator and full model separately. Operator runs sweep batches $32, 64$ with independent KV caches and decode 128 tokens. End-to-end runs match eight-GPU pipeline parallelism, batching (with unified batch size 24 and pipeline parallel size = 8 on Qwen), precision, KV layout, and generation settings; TPOT includes selection, index updates, sparse attention, communication, and pipeline waits. Initial index construction is reported separately, while incremental updates remain timed. Appendix C gives the full protocol. On a single GPU with a low batch size, FA2's flash decoding performs parallel computation over different temporal partitions of the key-value cache, so its observed speed may be close to that of SubCenSA. However, in this setting, neither HBM bandwidth nor Streaming Multiprocessors are fully utilized, making it impossible to provide a fair speed comparison that reflects the actual bottleneck determined by I/O plus computation. Thus we use 8-card, batch size 24 and pipeline parallel to perform end-to-end decoding to avoid such influences.

SubCenSA provides 4.104–$5.203\times$ operator speedup and **[TBD]**–**[TBD]**× end-to-end TPOT speedup. Initial index construction contributes **[TBD]** ms per request, while incremental updates account for **[TBD]**% of steady-state decoding time.

4.4 Design Ablations

On the independent development split, we ablate construction, temperature, sub-centroid count, and joint granularity/index density.

Table 6: RULER ablations (↑, 13-task macro-average). The construction panel uses a development comparison; the final panel jointly changes b, K_c, and summary type at matched selected-token density and is not a one-factor block-size ablation.

Index construction / Setting	**RULER**	**Temperature / Setting**	**RULER**
Max-norm + last	78.15	$\tau = 1$ (default)	**64.77**
Centered farthest-first + last radius-feasible (default)	**78.92**	$\tau = 0.1$	64.46
Centered + tolerant + last	77.85	$\tau \to 0^+$	64.31
K-means++	78.62		
Sub-centroids / Setting	**RULER**	**Granularity / Setting**	**RULER**
$K_c = 2$ (default)	64.77	$b = 32$, $K_c = 1$, mean	**65.23**
$K_c = 4$	**67.54**	$b = 64$, $K_c = 2$, temporal	65.08
$K_c = 8$	66.77	$b = 128$, $K_c = 4$, sub-centroid	63.38

Centered farthest-first is strongest, beating max-norm and k-means++ by 0.77 and 0.30 points; the mass-consistent $\tau = 1$ also performs best. Increasing K_c from 2 to 4 gains 2.77 points, while $K_c = 8$ falls 0.77 below $K_c = 4$. We retain $K_c = 2$ for lower metadata and scoring cost, using larger values only in equal-storage comparisons.

Because the granularity rows jointly vary b, K_c, and summary type, they show an end-to-end trade-off among selection granularity, index cost, and read amplification rather than an isolated block-size effect. Appendix G reports further aggregation and temperature analysis.

5 Conclusions

We introduced SubCenSA, a training-free selector for block-sparse decoding that replaces a single block representative with a small count-weighted mixture of sub-centroids. This simple but consequential design preserves both within-block heterogeneity and subcluster multiplicity while leaving the physical KV layout unchanged. The resulting estimator lower-bounds the true block mass and dominates mean-centroid pooling in exact arithmetic; for GQA, its residual structure further yields guarantees for shared top-m recovery, retained group mass, and weighted fixed-state output error. Combined with incremental index construction, block-aligned I/O, and exact selected-block attention, SubCenSA provides a principled path from compact block modeling to practical long-context decoding. Our evaluation connects this estimator-level improvement to selector fidelity, downstream quality, controlled design choices, and decoding efficiency.

Limitations and future work. SubCenSA retains the full KV cache, reducing decode-time KV traffic without reducing the cache memory footprint. The method is complementary to KV-cache quantization and compression, and combining them is a natural direction for reducing memory usage and read bandwidth together. SubCenSA's speedup compared with the dense softmax attention also demonstrates an observable the theotical upper-bound by I/O and computation, which could be narrowed through more sophisticated kernel optimization schemes.

References

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 (ACL)*, 2024. URL `https://arxiv.org/abs/2308.14508`.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3639–3664,

Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.183. URL https://aclanthology.org/2025.acl-long.183/.

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

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

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, pp. 35549–35562, 2024.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16344–16359, 2022.

Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *Proceedings of the First Conference on Language Modeling (COLM)*, 2024. URL https://arxiv.org/abs/2404.06654.

Junsung Hwang. LOCKS: Page-local compact key summaries for efficient long-context decoding. *arXiv preprint arXiv:2607.24555*, 2026.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2005.11401.

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

Di Liu, Meng Chen, Baotong Lu, Huiqiang Jiang, Zhenhua Han, Qianxi Zhang, Qi Chen, Chengruidong Zhang, Bailu Ding, Kai Zhang, Chen Chen, Fan Yang, Yuqing Yang, and Lili Qiu. RetrievalAttention: Accelerating long-context LLM inference via vector retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a. URL https://neurips.cc/virtual/2024/106433.

Guangda Liu, Chengwei Li, Jieru Zhao, Chenqi Zhang, and Minyi Guo. ClusterKV: Manipulating LLM KV cache in semantic space for recallable compression. *arXiv preprint arXiv:2412.03213*, 2024b. URL https://arxiv.org/abs/2412.03213.

Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, Zhiqi Huang, Huan Yuan, Suting Xu, Xinran Xu, Guokun Lai, Yanru Chen, Huabin Zheng, Junjie Yan, Jianlin Su, Yuxin Wu, Neo Y. Zhang, Zhilin Yang, Xinyu Zhou, Mingxing Zhang, and Jiezhong Qiu. MoBA: Mixture of block attention for long-context LLMs. In *Advances in Neural Information Processing Systems*, volume 38, 2025. Spotlight.

Daniel Ohayon, Itay Lamprecht, Itay Hubara, Israel Cohen, Daniel Soudry, and Noam Elata. Block sparse flash attention. *arXiv preprint arXiv:2512.07011*, 2025.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel R. Bowman. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2022. URL https://arxiv.org/abs/2112.08608.

David Reeb, Michael J. Kastoryano, and Michael M. Wolf. Hilbert's projective metric in quantum information theory. *Journal of Mathematical Physics*, 52(8):082201, 2011.

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*, volume 37, pp. 16692–16723. Curran Associates, Inc., 2024.

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

Alexander Tian, Aditya Ghai, Sanjit Neelam, Zaal Vasania, and Akshay Mishra. COBS: Cumulant order block sparse attention. *arXiv preprint arXiv:2607.09052*, 2026.

Georgios Tzachristas, Lei Deng, Ioannis Tzachristas, Gong Zhang, and Renhai Chen. A Mathematical Theory of Top-k Sparse Attention via Total Variation Distance. *arXiv preprint arXiv:2512.07647*, 2025.

Bailin Wang, Dan Friedman, Tao Lei, and Chong Wang. SPLA: Block sparse plus linear attention for long context modeling. *arXiv preprint arXiv:2601.22379*, 2026a.

Xinghao Wang, Pengyu Wang, Xiaoran Liu, Fangxu Liu, Jason Chu, Kai Song, and Xipeng Qiu. Prism: Spectral-aware block-sparse attention. In *Proceedings of the 43rd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2026b.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Yuxing Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23078–23097, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.1126. Best Paper Award.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17283–17297, 2020. URL https://arxiv.org/abs/2007.14062.

Fengji Zhang, Bei Chen, Yue Zhang, Jacky Keung, Jin Liu, Daoguang Zan, Yi Mao, Jian-Guang Lou, and Weizhu Chen. RepoCoder: Repository-level code completion through iterative retrieval and generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023a. URL https://arxiv.org/abs/2303.12570.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023b. URL https://arxiv.org/abs/2306.14048.

Weilin Zhao, Zihan Zhou, Zhou Su, Chaojun Xiao, Yuxuan Li, Yanghao Li, Yudi Zhang, Weilun Zhao, Zhen Li, Yuxiang Huang, Ao Sun, Xu Han, and Zhiyuan Liu. InfLLM-V2: Dense-sparse switchable attention for seamless short-to-long adaptation. *arXiv preprint arXiv:2509.24663*, 2025.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing of this manuscript, to support retrieval and discovery of related work, and to assist with research ideation and execution. We have not used generative AI tools to draft sections of the paper, to generate synthetic datasets, or to prove mathematical claims. We have reviewed all AI-assisted work: text polished with AI assistance was checked by the authors for technical accuracy, AI-assisted literature retrieval was verified against the original sources, and AI suggestions made during ideation and execution were evaluated and validated by the authors before being adopted. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

A SUB-CENTROID INDEX CONSTRUCTION

This appendix specifies the construction used for each completed key block. We refer to the frozen return rule throughout the paper as the *last radius-feasible snapshot*: among the initial snapshot and two subsequent Lloyd snapshots, it returns the latest snapshot whose covering radius does not exceed that of the initial snapshot. Initialization and snapshot-return rules are selected on the independent development split; the same construction is then held fixed for the reported test settings.

A.1 FROZEN CONSTRUCTION RULE

Let $\mathbf{K} \in \mathbb{R}^{N_{\mathrm{blk}} \times b \times d_h}$ contain the complete key blocks. Construction is independent across blocks and KV heads, but all operations are batched in the implementation. We assume $1 \le K_c \le b$. For a partition $\mathcal{P} = \{\mathcal{C}_1, \dots, \mathcal{C}_{K_c}\}$, define

$$W(\mathcal{P}) = \sum_{j=1}^{K_c} \sum_{i \in \mathcal{C}_j} \|\mathbf{k}_i - \mathbf{c}_j\|_2^2, \qquad D(\mathcal{P}) = \max_j \max_{i \in \mathcal{C}_j} \|\mathbf{k}_i - \mathbf{c}_j\|_2. \tag{19}$$

Here W is the within-cluster distortion and D is the covering radius. Lloyd updates monotonically reduce W; no monotonicity claim is made for D. The covering-radius condition is an additional snapshot safeguard.

For the endpoint branches $K_c = 1$ and $K_c = b$, we regard the returned partition as all three snapshots. Thus W_t and D_t are constant over the notional trajectory, and the statements below continue to apply.

A.2 TIE-BREAKING AND EMPTY-CLUSTER REPAIR

All exact ties are resolved deterministically. Seed ties are broken by the smallest token index. In the initial assignment, each seed token owns its seed cluster on a zero-distance tie; all remaining ties use the smallest seed-cluster identifier. During Lloyd assignment, a key keeps its previous cluster whenever that cluster is among the tied minimizers; otherwise, it takes the smallest tied cluster identifier. Canonicalizing cluster labels by the smallest member token index also remaps the stored assignments before the next pass.

Empty clusters are processed in ascending cluster order. A donor key must belong to a cluster containing at least two keys. Among eligible donors, the key with the largest squared residual to its current centroid is moved to the empty cluster, with ties broken by token index. The donor becomes a temporary singleton centroid, counts are updated immediately, and all centroids are subsequently recomputed as exact empirical means. Since $K_c \le b$, a donor is available whenever an empty cluster exists.

A.3 CONSTRUCTION GUARANTEES

Let $R^\star_{K_c}(B)$ denote the optimal covering radius of block B using K_c arbitrary centers.

Proposition 4 (Construction guarantees)**.** *For every block, Algorithm 1 satisfies:*

1. *the Lloyd trajectory has non-increasing distortion, $W_{t+1} \le W_t$;*

Algorithm 1 Sub-centroid construction with the last radius-feasible snapshot rule

Require: Key blocks $\mathbf{K} \in \mathbb{R}^{N_{\mathrm{blk}} \times b \times d_h}$, sub-centroid count K_c, total snapshots $T_c = 3$
Ensure: Centroids $\mathbf{C} \in \mathbb{R}^{N_{\mathrm{blk}} \times K_c \times d_h}$, counts $\mathbf{n} \in \mathbb{N}^{N_{\mathrm{blk}} \times K_c}$

Apply all steps independently to every block in FP32 working precision.
if $K_c = 1$ **then**
  Return the block mean with count b.
else if $K_c = b$ **then**
  Return the original keys with unit counts.
end if
Compute $\boldsymbol{\mu}_B = b^{-1} \sum_{i=1}^{b} \mathbf{k}_{B,i}$.
Select $i_{B,1} = \arg\max_i \|\mathbf{k}_{B,i} - \boldsymbol{\mu}_B\|_2^2$.
for $j = 2, \ldots, K_c$ **do**
  Select the unused key maximizing its distance to the nearest previously selected seed.
end for
Assign each key to its nearest seed.
Repair empty clusters, recompute empirical centroids and counts, and canonicalize cluster labels.
Save the initial snapshot $\mathcal{P}_0$ and compute $W_0 = W(\mathcal{P}_0)$ and $D_0 = D(\mathcal{P}_0)$.
For each block, set $\mathcal{P}_{\mathrm{return}} \leftarrow \mathcal{P}_0$.
for $t = 1, \ldots, T_c - 1$ **do**
  Assign each key to its nearest current centroid.
  Repair empty clusters, recompute empirical centroids and counts, and canonicalize cluster labels.
  Form $\mathcal{P}_t$ and compute W_t, D_t separately for every block.
  if $D_t \le D_0$ for a block **then**
    Set that block's $\mathcal{P}_{\mathrm{return}} \leftarrow \mathcal{P}_t$.
  end if
  Continue the Lloyd trajectory from $\mathcal{P}_t$ regardless of whether $\mathcal{P}_t$ is radius-feasible.
end for
Return the per-block snapshots $\mathcal{P}_{\mathrm{return}}$.

2. *the count-weighted between-cluster scatter* $\sum_j n_j \|\mathbf{c}_j - \boldsymbol{\mu}_B\|_2^2$ *is non-decreasing;*

3. *the initial snapshot satisfies* $D_0 \le 4R^\star_{K_c}(B)$*;*

4. *the returned snapshot* $\mathcal{P}_{\mathrm{return}}$ *satisfies* $D(\mathcal{P}_{\mathrm{return}}) \le D_0$ *and hence*

$$e_B \le 8\|\mathbf{a}\|_2^2 \big(R^\star_{K_c}(B)\big)^2. \tag{20}$$

Proof. Nearest-centroid assignment cannot increase distortion relative to the current centroids. Empty-cluster repair moves a donor to a singleton cluster, reducing its residual to zero before the mean update. Replacing each centroid by its empirical mean minimizes the within-cluster squared-error objective. Thus $W_{t+1} \le W_t$.

For exact empirical means,

$$\sum_{i \in B} \|\mathbf{k}_i - \boldsymbol{\mu}_B\|_2^2 = W_t + \sum_j n_j \|\mathbf{c}_j - \boldsymbol{\mu}_B\|_2^2.$$

The left-hand side is independent of the partition, so each decrease in W_t gives an equal increase in the between-cluster scatter.

The first centered seed is itself a data point, after which the standard farthest-first rule applies. Its seed covering radius therefore satisfies $D_{\mathrm{seed}} \le 2R^\star_{K_c}(B)$ (Gonzalez, 1985). After initial assignment, every key lies within D_{seed} of its seed. Repair only partitions assigned sets further, and every point in a subset of such a ball is within $2D_{\mathrm{seed}}$ of the subset mean. Hence $D_0 \le 2D_{\mathrm{seed}} \le 4R^\star_{K_c}(B)$. The return rule ensures $D(\mathcal{P}_{\mathrm{return}}) \le D_0$; substituting this inequality into Eq. equation 10 proves Eq. equation 20. □

Table 7: SubCenSA decode-time kernels.

Kernel	Parallelization	Role
Index construction	KV heads × completed blocks	Build new block summaries
Normalizer scan	KV-head groups	Accumulate $\widehat{Z}_h$
Score–top-m scan	KV-head groups	Form $g_G(B)$ and select blocks
Split score–merge	Block-dimension slabs	Improve low-batch occupancy
Batched GEMM scoring	Batched head groups	Materialized high-occupancy path
Sparse attention	KV head × value tile	Attend to selected original KV

The proposition separates two properties: Lloyd updates reduce W, whereas the snapshot rule controls D. Neither property implies monotonic improvement of the realized query-dependent residual, and the successive Lloyd partitions need not be nested.

A.4 Complexity and Storage

The batched construction costs $O(N_{\mathrm{blk}} b K_c d_h T_c)$ operations, with small fixed K_c and T_c. Newly completed blocks are appended incrementally. If centroid components use s_c bytes and counts use s_n bytes, the persistent metadata per block and KV head is

$$S_{\mathrm{SC}}(K_c) = K_c(d_h s_c + s_n) \quad \text{bytes.} \tag{21}$$

For $K_c = 2$, $b = 64$, FP16 centroids, and uint16 counts, this is approximately 1.6% of the corresponding FP16 key–value payload, excluding temporary buffers and allocator padding. Quantization is applied only after snapshot selection.

At query time, the temperature-family score used in the ablations is

$$\widehat{L}^{(\tau)}(\mathbf{q}, B) = \tau \log \sum_{j=1}^{K_c} n_{B,j} \exp\left(\frac{\mathbf{a}^\top \mathbf{c}_{B,j}}{\tau}\right). \tag{22}$$

The frozen default is $\tau = 1$, which is exactly $\widehat{L}_{\mathrm{SC}}$ in Eq. equation 8.

B Decode-Time Kernel Implementation

Let N_{blk} be the number of complete blocks, H_{kv} the number of KV heads, and $r = H_q/H_{\mathrm{kv}}$ the number of query heads sharing one KV head.

B.1 Index Layout and Candidate Conventions

Centroids are stored as

$$\mathbf{C} \in \mathbb{R}^{H_{\mathrm{kv}} \times d_h \times (N_{\mathrm{blk}} K_c)},$$

with the K_c centroids of adjacent blocks contiguous in the scoring layout. Counts are represented by their log values $\log n_{B,j}$. Mandatory complete blocks are excluded from optional candidates. The incomplete tail is not indexed; it belongs to the exact mandatory atom and is read directly. Invalid or padded blocks receive score $-\infty$. Score ties are resolved by smaller physical block identifier.

For $m > 0$, the online selector uses an internal capacity $S = 2^{\lceil \log_2 m \rceil} \geq m$. Each tile retains at least its top S candidates before merging with the global buffer, so truncating the final buffer to its first m entries gives the exact top-m. When $m = 0$, the selector returns the mandatory identifiers directly. Optional identifiers are kept in score order; contiguity refers to the b entries within each selected block.

B.2 Kernel Summary

B.3 Default Two-Pass Normalized Selector

No candidate pruning occurs before the first-pass normalizers are complete. For centroid bytes s_c and count bytes s_n, the logical index size per KV head is $S_{\mathrm{index}} = N_{\mathrm{blk}} K_c(d_h s_c + s_n)$; the default

Algorithm 2 Default two-pass self-normalized selector

Require: Query heads $\{\mathbf{q}_h\}_{h\in G}$, compact index, mandatory set $\mathcal{A}$, optional budget m, weights $\{\beta_h\}$
Ensure: Mandatory and selected optional block identifiers
if $m = 0$ **then**
  Return the mandatory identifiers.
end if
for $h \in G$ **do**
  Compute exact mandatory mass $M_h(\mathcal{A})$.
  Initialize $\log \widehat{Z}_h \leftarrow \log M_h(\mathcal{A})$.
end for
for each optional block tile $\mathcal{T}$ **do**
  for $h \in G$ and $B \in \mathcal{T}$ **do**
    Compute $\widehat{L}_{h,B} = \mathrm{LSE}_j(\mathbf{a}_h^\top \mathbf{c}_{B,j} + \log n_{B,j})$.
  end for
  Accumulate the tile masses into $\log \widehat{Z}_h$ for every head.
end for
Initialize an empty top-S buffer, $S = 2^{\lceil \log_2 m \rceil}$.
for each optional block tile $\mathcal{T}$ **do**
  Recompute $\widehat{L}_{h,B}$.
  Form $g_G(B) = \mathrm{LSE}_{h\in G}[\widehat{L}_{h,B} - \log \widehat{Z}_h + \log \beta_h]$.
  Set invalid and mandatory candidates to $-\infty$.
  Merge the tile candidates into the top-S buffer using score-descending, block-id-ascending order.
end for
Return mandatory identifiers followed by the first m optional identifiers.

selector performs two complete logical scans before cache reuse and program-level physical traffic are considered.

B.4 Alternative Scoring Paths and Selected-Block Attention

A one-pass raw-mass ablation replaces the normalized score by $\Gamma_B = \mathrm{LSE}_{h\in G}[\widehat{L}_{h,B} + \log \gamma_h]$. It estimates a different objective and is not the default selector. The fused and split paths maintain online top-m state without materializing a full group-score matrix. At low occupancy, the block dimension is divided into disjoint slabs; each slab returns a local top-S list and a second kernel exactly merges the lists. The framework-reference and batched-GEMM paths instead materialize query–centroid or block-score tensors.

After selection, one Triton program per KV head and value tile performs online-softmax attention over the mandatory and selected original KV blocks. Within a GQA group, a selected block is reused across its query heads. Causal, tail, and invalid-position masks are applied before the online-softmax update, and padded identifiers are set to -1 and skipped. The only algorithmic approximation in this stage is block selection.

C Experimental Details

This section records the protocol shared by the main-text experiments. It intentionally omits unrun appendix-only experiments and does not infer values absent from the frozen run manifests.

C.1 Models, Benchmarks, and Decoding

The main RULER grid uses the Qwen3-30B-A3B, Qwen3-8B, and Qwen3.5-9B checkpoints named in Table 4; Qwen3.5-4B is used for controlled ablations and profiling. LongBench and LongBench-V2-CoT use Qwen3-30B-A3B. For hybrid-attention Qwen3.5 models, SubCenSA replaces only

Table 8: Main benchmark configurations. Exact checkpoint revisions and record identifiers are determined by the frozen manifests.

Benchmark	Model	Nominal KV budget
RULER 64K	Qwen3-30B/Qwen3-8B/Qwen3.5-9B	4,096
RULER 128K	Qwen3-30B/Qwen3-8B/Qwen3.5-9B	8,192
RULER 256K	Qwen3-30B/Qwen3.5-9B	16,384
LongBench	Qwen3-30B-A3B-Instruct-2507	15%
LongBench-V2-CoT	Qwen3-30B-A3B-Instruct-2507	1,024

the full-softmax computation: linear-attention states, learned projections, normalization, and gates remain unchanged, and the reported KV-read fraction concerns the indexed full-attention cache only.

All indices are constructed from the transformed keys actually stored in the KV cache, after model-specific normalization and positional encoding. Centroids are not normalized again. The scale $1/\sqrt{d_h}$ is applied once in both query–key and query–centroid scores.

RULER. All 13 configured tasks are used. Each reported setting uses a frozen record subset shared across methods and budgets. The prompt is the stored context followed by its question, a newline, and its answer prefix, with the per-record generation limit and no additional chat template. The score of one example is the fraction of its gold answer strings appearing in the generated text under case-insensitive substring matching; the overall score is the macro-average over the 13 tasks on a 0–100 scale.

LongBench. The 16 English tasks are used, capped at the first 100 records per task, with the stored task templates and no additional chat template. A filled prompt longer than 32,768 tokens keeps its first and last 16,384 tokens, and generation limits follow the stored benchmark metadata. Scoring uses the benchmark's standard per-task metrics, taking the maximum over reference answers with the evaluator's standard preprocessing; the final score is a macro-average of task means.

LongBench-V2-CoT. All 503 selected records are evaluated with the stored CoT multiple-choice template and up to 2,048 generated tokens. The nominal prompt cap of 131,072 tokens, together with 2,048 reserved output positions and 64 positions of margin, caps the final input at 128,960 tokens; overlong inputs retain the beginning and end. The answer extractor uses the last explicit answer declaration when available and otherwise the last standalone A–D option letter.

Decoding settings. Generation is greedy with one beam, the checkpoint-specific end-of-sequence condition, and the benchmark-specific generation cap. Prompt prefilling is dense; sparse attention begins with subsequent single-token decode calls, so the first generated token uses the dense prefill output. Index state is cleared between prompts. Failed jobs are recorded as failed rather than replaced with fewer examples, shorter contexts, or different budgets.

C.2 Read-Budget Accounting

Let T be the current valid KV length, $b = 64$, $N = \lfloor T/b \rfloor$, and $r = T - Nb$. Only the N complete blocks are indexed, and inputs with fewer than two complete blocks use the dense fallback. On the sparse path, a fixed nominal budget B_{nom} gives

$$k_{\mathrm{full}} = \min\left\{N, \max\left(1, \left\lfloor \frac{B_{\mathrm{nom}}}{b} \right\rfloor\right)\right\}, \tag{23}$$

and a fractional budget f gives

$$k_{\mathrm{full}} = \max\{1, \min(N, \lceil fN \rceil)\}. \tag{24}$$

The first complete block receives mandatory priority, followed by the latest complete block whenever at least two complete-block slots are available; overlapping identifiers count once. Thus the mandatory set is formed within the allowance and the optional budget $m = k_{\mathrm{full}} - |\mathcal{A}_{\mathrm{full}}|$ is nonnegative. When $r > 0$, the unindexed partial tail is read in addition to the nominal complete-block

budget, so $T_{\text{read}} = bk_{\text{full}} + r$. Thus the reported KV budget excludes at most 63 additional valid tail tokens and does not denote cache-capacity reduction. Token-read fractions also exclude index scans, temporary buffers, and repeated physical loads across value tiles.

The frozen default runs Algorithm 1 with $b = 64$, $K_c = 2$, $T_c = 3$, $\tau = 1$, and the self-normalized group score of Eq. equation 13. Completed blocks are built in batches at the first sparse decode call and appended incrementally; the partial tail remains directly readable until it becomes complete.

C.3 Numerics and Selector Diagnostics

The accuracy path accumulates log-sum-exp and online-softmax reductions in FP32. Fused entry points may use reduced-precision centroid views with FP32 accumulation. These are implementation statements, not certified all-input numerical bounds; Appendix G.3 gives a conditional perturbation analysis.

Diagnostics are collected from realized sparse trajectories. For a sample x with true top-m_x optional set $S_x^\star$ and estimated set $\widehat{S}_x$,

$$\text{Recall@}\, m_x = \frac{|S_x^\star \cap \widehat{S}_x|}{m_x}, \qquad \text{MassRatio}_{\text{opt}} = \frac{J_x(\widehat{S}_x)}{J_x(S_x^\star)},$$

$$\text{MassRatio}_{\text{total}} = \frac{J_x(\mathcal{A}_x) + J_x(\widehat{S}_x)}{J_x(\mathcal{A}_x) + J_x(S_x^\star)}.$$

Mass-NDCG uses $J_x(B)$ as relevance with logarithmic rank discount. Every diagnostic uses the same head aggregation as the evaluated selector.

C.4 Accuracy and Profiling Scope

The elapsed time recorded by the accuracy runner includes prefill, generation, and diagnostic collection and is not used as a latency measurement. Operator profiling measures one attention layer with synthetic tensors matching the target model shape, independent KV caches for every batch element, a batch sweep of $1, 4, 8, 16, 32, 64$, and 128 decode tokens. End-to-end profiling measures the complete model and includes all layers, KV append, selection, sparse attention, communication, pipeline wait, and sampling. Initial prompt-index construction is reported separately; incremental updates remain in the steady-state decode interval.

C.5 Execution and Reproducibility

The frozen manifests record example identifiers, tokenized-input hashes, checkpoint and tokenizer revisions, model configurations, software versions, code fingerprints, generated text, scores, and diagnostic paths. Resume rejects mismatched manifests or fingerprints. Missing or failed examples are not replaced, and aggregation reconstructs benchmark tasks rather than averaging worker shards equally.

D Baseline Adaptations and Storage Accounting

D.1 Common Wrapper and Naming

The common wrapper changes only the persistent block summary and scoring rule. Every contiguous-block adapter retains the original KV cache, uses the same self-normalized GQA aggregation and block budget, and invokes the same exact selected-block attention backend. Names ending in "-Index" refer to the adapted selector statistic, not to a complete native sparse- attention system.

D.2 Scoring Rules

For one head, let $\mathbf{a}_h = \mathbf{q}_h/\sqrt{d_h}$ and $\boldsymbol{\mu}_B = b^{-1}\sum_{i\in B}\mathbf{k}_i$.

Table 9: Per-head scores of the contiguous-block adapters.

Method	Persistent summary	Per-head log score
Centroid-Index	$\boldsymbol{\mu}_B$	$\log b + \mathbf{a}_h^\top \boldsymbol{\mu}_B$
Quest-Index	coordinate minima and maxima	$\log b + \sum_\ell \max\{a_{h\ell} k_{B\ell}^{\min}, a_{h\ell} k_{B\ell}^{\max}\}$
SPLA-Index	mean and coordinate variance	$\log b + \mathbf{a}_h^\top \boldsymbol{\mu}_B + \log(1 + u_{hB})$
COBS-Index	mean and compressed covariance factors	$\log b + \mathbf{a}_h^\top \boldsymbol{\mu}_B + \frac{1}{2} \mathbf{a}_h^\top \widehat{\boldsymbol{\Sigma}}_B \mathbf{a}_h$
LOCKS-Index	page-local spectral factors	log-sum-exp of reconstructed within-block logits
SubCenSA	K_c centroids and counts	$\mathrm{LSE}_j(\mathbf{a}_h^\top \mathbf{c}_{B,j} + \log n_{B,j})$

Here $u_{hB} = \frac{1}{2} \sum_\ell a_{h\ell}^2 v_{B\ell}$. The Quest score is an axis-aligned box upper proxy; coordinate extrema need not belong to the same key. In contrast, SubCenSA is a one-sided lower mass estimate in exact arithmetic. Methods are therefore compared by ranking, retained mass, and downstream quality rather than by treating all score errors as having the same sign.

D.3 Adapter Scope

Centroid-Index is empirical mean pooling rather than a complete learned routing architecture. Quest-Index uses the coordinate-bound statistic in the common wrapper rather than the complete native serving system. SPLA-Index omits the native residual linear-attention branch and associated training procedure. COBS-Index uses only the adapted compressed second-order block score rather than the native NSA-based architecture and covariance-compression pipeline. LOCKS-Index denotes the adapted spectral summary and does not imply reproduction of the complete native quantization, page-factorization, serving-plugin, CUDA-graph, or kernel stack. Prism is discussed as a spectral-aware prefill method; any decode score derived from it is an adapter rather than a native Prism decode system.

D.4 Storage and Cost Scope

For the contiguous indices, nominal vector counts are one for Centroid, two for Quest and SPLA-Index, and K_c centroids plus K_c counts for SubCenSA. SubCenSA stores $K_c(d_h s_c + s_n)$ bytes per complete block and KV head. Exact packed-layout comparisons must additionally count each method's scales, eigenvalues, and alignment padding. Allocator padding, temporary layout conversions, score tensors, and selection buffers are separate costs. Equal vector counts or nominal token budgets therefore do not by themselves imply equal metadata bytes, memory traffic, or selector latency.

E Estimator and Selection Analysis

This section proves the results stated in Sections 2–3 in the order of the theory chain, restating in full those auxiliary results that the main text summarizes informally: the output-error target (Theorem 5 and Corollary 6), the Jensen hierarchy and its KL identity (Theorem 1 and Lemma 8), the between/within characterization (Theorem 9), the one-sided ranking guarantees (Theorem 12), and the self-normalized GQA selection guarantees (Theorems 16 and 17), together with the interval certificate of Theorem 14. Appendix F establishes the necessity of the count weights; Appendix G analyzes the design choices (grouped-query aggregation, the temperature family, and finite-precision robustness).

E.1 Notation and Standing Definitions

Notation follows Sections 3.1 and 3.2; the standing symbols are collected in the table below, where D_t, W_t denote $D(\mathcal{P}), W(\mathcal{P})$ at the snapshot P_t of Algorithm 1, and τ_G is unrelated to the score temperature τ of Appendix G.2.

Symbol	Definition
M_B, L_B	block mass $M_B = \sum_{i\in B} e^{\mathbf{a}^\top \mathbf{k}_i}$ and $L_B = \log M_B$
$s_{h,i}$, $\mathbf{a}_h$	per-head score $s_{h,i} = \mathbf{a}_h^\top \mathbf{k}_i$ with $\mathbf{a}_h = \mathbf{q}_h/\sqrt{d_h}$
$M_h(B)$, Z_h, $p_h(B)$	head mass $\sum_{i\in B} e^{s_{h,i}}$, normalizer over all valid tokens, $p_h = M_h/Z_h$
$\mathcal{A}$, $\mathcal{O}$	mandatory set and optional blocks, viewed as *atoms*
e_x	atom log-residual $\log M_x - \log \widehat{M}_x$ (of either sign; ≥ 0 for one-sided lower estimates)
ω_h, Ω, τ_G	oscillation $\max_x e_x - \min_x e_x$; $\Omega = \max_{h\in G} \omega_h$; $\tau_G = \sum_{h\in G} \beta_h \tanh(\omega_h/4)$, $\beta_h \geq 0$, $\sum_h \beta_h = 1$
$D(\mathcal{P})$, $W(\mathcal{P})$	covering radius $\max_j \max_{i\in\mathcal{C}_j} \|\mathbf{k}_i - \mathbf{c}_j\|_2$; distortion $\sum_j \sum_{i\in\mathcal{C}_j} \|\mathbf{k}_i - \mathbf{c}_j\|_2^2$
R_B	block radius $\max_{i\in B} \|\mathbf{k}_i - \boldsymbol{\mu}_B\|_2$

The non-emptiness of the subclusters costs no generality: Algorithm 1 enforces non-emptiness through donor-point repair, with zero-count clusters masked as a numerical safeguard, so only non-empty subclusters enter Eq. equation 8, each represented by its exact empirical mean.

E.2 Output-Error Target and the GQA Proxy

Theorem 5 (Truncation error of sparse reads: exact decomposition and minimax optimality)**.** *Let R be a non-empty read set of tokens, let $Z = \sum_i \exp(s_i)$ over all valid tokens, and let $\alpha(R) = \big(\sum_{i\in R} \exp(s_i)\big)/Z$. With $\mathbf{o}$ the dense output, $\mathbf{o}_R$ Eq. equation 1 with $\mathcal{R}(\mathbf{q}) = R$, and $\mathbf{o}_{\bar{R}}$ the output restricted to the complement $\bar{R}$ (when non-empty),*

$$\mathbf{o} - \mathbf{o}_R = \big(1 - \alpha(R)\big)\big(\mathbf{o}_{\bar{R}} - \mathbf{o}_R\big), \tag{25a}$$

$$\|\mathbf{o} - \mathbf{o}_R\|_2 \leq \big(1 - \alpha(R)\big) \max_{i\in R,\, j\notin R} \|\mathbf{v}_i - \mathbf{v}_j\|_2 \;\leq\; 2V\big(1 - \alpha(R)\big), \tag{25b}$$

the second line holding whenever $\|\mathbf{v}_i\|_2 \leq V$ for all tokens. The last bound is tight in the value-agnostic worst case—attained by antipodal values $\mathbf{v}_i = \mp V\mathbf{u}$ for a unit vector $\mathbf{u}$ (proof below). Consequently, with the mandatory set, the budget m, and equal-size optional blocks fixed, selecting the m optional blocks with the largest true masses $M(\mathbf{q}, B)$ is minimax optimal for this worst-case output error among all deterministic value-agnostic block-selection rules.

Proof of Theorem 5. With $Z_R = \sum_{i\in R} e^{s_i}$, $\alpha(R) = Z_R/Z$, $Z_{\bar{R}} = Z - Z_R$, and $\mathbf{o}_{\bar{R}} = Z_{\bar{R}}^{-1} \sum_{j\notin R} e^{s_j}\mathbf{v}_j$ (when $Z_{\bar{R}} > 0$), the dense output is the convex combination $\mathbf{o} = \alpha(R)\mathbf{o}_R + (1 - \alpha(R))\mathbf{o}_{\bar{R}}$, which rearranges to Eq. equation 25a. Both $\mathbf{o}_R$ and $\mathbf{o}_{\bar{R}}$ are convex combinations of value vectors over disjoint token sets, so

$$\|\mathbf{o}_{\bar{R}} - \mathbf{o}_R\|_2 \leq \max_{i\in R,\, j\notin R} \|\mathbf{v}_i - \mathbf{v}_j\|_2 \leq 2V,$$

proving Eq. equation 25b. For tightness, take $\mathbf{v}_i = -V\mathbf{u}$ on R and $\mathbf{v}_i = V\mathbf{u}$ on the complement for a unit vector $\mathbf{u}$: then $\|\mathbf{o} - \mathbf{o}_R\|_2 = (1 - \alpha(R)) \cdot 2V$; if $\bar{R}$ is empty, $\alpha(R) = 1$ and all three displays vanish. Finally, for any selection S of m optional blocks,

$$\alpha(\mathcal{A} \cup S) = \alpha(\mathcal{A}) + \sum_{B\in S} \frac{M(\mathbf{q}, B)}{Z},$$

so the worst-case bound $2V(1 - \alpha(\mathcal{A} \cup S))$ is minimized by maximizing $\sum_{B\in S} M(\mathbf{q}, B)$, i.e., by selecting the m optional blocks with the largest true masses. □

Corollary 6 (Output-error proxy for a shared GQA read set)**.** *Let G be a group of query heads sharing one KV head, with per-head retained probabilities $\alpha_h(R) = \sum_{i\in R} p_{h,i}$, weights $\beta_h \geq 0$ with $\sum_{h\in G} \beta_h = 1$, and a common value-norm bound V. With the group objective $J_G(R) = \sum_{h\in G} \beta_h \alpha_h(R)$, every optional selection S satisfies*

$$\sum_{h\in G} \beta_h \|\mathbf{o}_h - \mathbf{o}_{h,\mathcal{A}\cup S}\|_2 \leq 2V\Big[1 - J_G(\mathcal{A}) - \sum_{B\in S} J_G(B)\Big], \tag{26}$$

so ranking optional blocks by their group score $J_G(B)$ and keeping the top-m minimizes this shared-set proxy bound among deterministic selections. We adopt the uniform default $\beta_h = 1/|G|$, i.e., the average head-level probability retention.

Proof of Corollary 6. Apply Eq. equation 25b of Theorem 5 to each head with $R = \mathcal{A} \cup S$, weight by β_h, and sum; among selections of fixed size m only $\sum_{B \in S} J_G(B)$ depends on S, so the bound is minimized by the top-m blocks under J_G. □

E.3 Proof of Theorem 1 and Corollary 7; KL Identity

Proof of Theorem 1. Write $z_i = \mathbf{a}^\top(\mathbf{k}_i - \mathbf{c}_j)$ for $i \in \mathcal{C}_j$, so $\sum_{i \in \mathcal{C}_j} z_i = 0$. Jensen's inequality gives $\frac{1}{n_j} \sum_{i \in \mathcal{C}_j} e^{z_i} \geq e^{\frac{1}{n_j} \sum_i z_i} = 1$ within every subcluster, and therefore

$$M_B = \sum_{j=1}^{K_c} e^{\mathbf{a}^\top \mathbf{c}_j} \sum_{i \in \mathcal{C}_j} e^{z_i} \geq \sum_{j=1}^{K_c} n_j e^{\mathbf{a}^\top \mathbf{c}_j} = \widehat{M}_{\mathcal{P}},$$

the right inequality. For the left one, $\boldsymbol{\mu}_B = \sum_{j=1}^{K_c} \frac{n_j}{b} \mathbf{c}_j$ and the convexity of the exponential give $e^{\mathbf{a}^\top \boldsymbol{\mu}_B} = \exp(\sum_j \frac{n_j}{b} \mathbf{a}^\top \mathbf{c}_j) \leq \sum_j \frac{n_j}{b} e^{\mathbf{a}^\top \mathbf{c}_j}$, and multiplying by b yields $\widehat{M}_{\text{mean}} \leq \widehat{M}_{\mathcal{P}}$. This step uses only that each $\mathbf{c}_j$ is the mean of its own subcluster, so it holds for every non-empty in-block partition.

For refinement monotonicity it suffices to split one parent cluster C into children $C_1, \ldots, C_r$: the parent centroid is the count-weighted mean of the children's centroids, $\mathbf{c} = \sum_{\ell=1}^{r} \frac{n_\ell}{n} \mathbf{c}_\ell$, so the same convexity argument gives $n\, e^{\mathbf{a}^\top \mathbf{c}} \leq \sum_{\ell=1}^{r} n_\ell e^{\mathbf{a}^\top \mathbf{c}_\ell}$. Repeating over all refined clusters yields $\widehat{M}_{\mathcal{P}} \leq \widehat{M}_{\mathcal{P}'}$, and the upper bound $\widehat{M}_{\mathcal{P}'} \leq M$ is the cluster-wise Jensen bound from the first paragraph. The endpoints $K_c = 1$ and $K_c = b$ are immediate from the definitions.

For the residual bound, convexity on $[-r_j, r_j]$ and the zero mean of z_i give $\frac{1}{n_j} \sum_{i \in \mathcal{C}_j} e^{z_i} \leq \cosh r_j \leq e^{r_j^2/2}$. Hence

$$\frac{M_B}{\widehat{M}_{\text{SC}}} \leq \sum_j \pi_j e^{r_j^2/2} \leq e^{r_B^2/2}.$$

Taking logarithms gives the first three inequalities in Eq. equation 10; the last follows from $r_B \leq \|\mathbf{a}\|_2 D_B$.

For the equality conditions: the exponential is strictly convex, so the cluster-wise Jensen step is an equality if and only if all z_i inside the subcluster coincide; since their mean is zero, this means $\mathbf{a}^\top \mathbf{k}_i$ is constant within every subcluster. The left inequality is an equality if and only if all $\mathbf{a}^\top \mathbf{c}_j$ coincide, again by strict convexity. □

Corollary 7 (Pointwise dominance over mean pooling)**.** *For every query and block,*

$$0 \leq e_B^{\text{SC}} := L(\mathbf{q}, B) - \widehat{L}_{\text{SC}}(\mathbf{q}, B) \leq L(\mathbf{q}, B) - \widehat{L}_{\text{mean}}(\mathbf{q}, B),$$

and the same dominance holds for the absolute mass error. This is a statement about the one-sided mass estimates only: it does not by itself imply smaller errors of the normalized *block probabilities, fewer ranking inversions, or better downstream accuracy (Proposition 15).*

Proof of Corollary 7. Both estimators lower-bound M_B, and $\widehat{M}_{\text{mean}} \leq \widehat{M}_{\text{SC}} \leq M_B$ by Theorem 1; subtracting and applying the increasing logarithm gives both the log-residual and the absolute mass-error dominance. □

The log-mass residual has an exact information-theoretic meaning, which clarifies what the estimator does and does not preserve.

Lemma 8 (KL identity for the log-mass residual)**.** *Within a block, let $p_i = e^{\mathbf{a}^\top \mathbf{k}_i}/M_B$ be the exact attention weights and let $\widetilde{p}_i = e^{\mathbf{a}^\top \mathbf{c}_j}/\widehat{M}_{\text{SC}}$ for $i \in \mathcal{C}_j$ be the coreset weights. Then*

$$e_B^{\text{SC}} = L_B - \widehat{L}_{\text{SC}} = D_{\text{KL}}(\widetilde{p} \,\|\, p).$$

Proof of Lemma 8. Write $z_i = \mathbf{a}^\top(\mathbf{k}_i - \mathbf{c}_j)$ and recall $\sum_{i\in\mathcal{C}_j} z_i = 0$. For $i \in \mathcal{C}_j$,

$$\log \frac{\widetilde{p}_i}{p_i} = \mathbf{a}^\top \mathbf{c}_j - \log \widehat{M}_{\mathrm{SC}} - \mathbf{a}^\top \mathbf{k}_i + \log M_B = e_B^{\mathrm{SC}} - z_i.$$

Therefore

$$D_{\mathrm{KL}}(\widetilde{p}\|p) = \sum_i \widetilde{p}_i\big(e_B^{\mathrm{SC}} - z_i\big) = e_B^{\mathrm{SC}} - \sum_j \frac{n_j e^{\mathbf{a}^\top \mathbf{c}_j}}{\widehat{M}_{\mathrm{SC}}} \cdot \frac{1}{n_j} \sum_{i\in\mathcal{C}_j} z_i = e_B^{\mathrm{SC}},$$

since the inner sum vanishes for every j. □

Lemma 8 shows that SubCenSA keeps the differences *between* subclusters but flattens the logits *within* each subcluster to a single representative value, and that its log-mass residual is exactly the KL divergence incurred by this flattening.

E.4 Between-Cluster Gain and Within-Cluster Residual

For a bounded zero-mean variable with $|X| \le r$, convexity gives $\mathbb{E}[e^X] \le \cosh r$, and Hoeffding's lemma gives $\cosh r \le e^{r^2/2}$ (Hoeffding, 1963).

Theorem 9 (Between-cluster gain and within-cluster residual)**.** *Let $w_j = n_j/b$ and $\boldsymbol{\epsilon}_i = \mathbf{k}_i - \mathbf{c}_j$ for $i \in \mathcal{C}_j$; write $\boldsymbol{\Sigma}_j = \frac{1}{n_j}\sum_{i\in\mathcal{C}_j} \boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_i^\top$, $\boldsymbol{\Sigma}_{\mathrm{between}} = \sum_j w_j(\mathbf{c}_j - \boldsymbol{\mu}_B)(\mathbf{c}_j - \boldsymbol{\mu}_B)^\top$, and $\boldsymbol{\Sigma}_{\mathrm{within}} = \sum_j w_j \boldsymbol{\Sigma}_j$.*

(A) *With $Y_j = \mathbf{a}^\top(\mathbf{c}_j - \boldsymbol{\mu}_B)$ and $R_{\mathrm{bet}} = \max_j |Y_j|$, the gain over mean pooling is exactly $\widehat{L}_{\mathrm{SC}} - \widehat{L}_{\mathrm{mean}} = \log\sum_j w_j e^{Y_j}$, is zero if and only if all Y_j vanish, and is at most $\frac{1}{2}R_{\mathrm{bet}}^2$ (the two-sided variance form appears in the proof below).*

(B) *With $z_i = \mathbf{a}^\top(\mathbf{k}_i - \mathbf{c}_j)$ for $i \in \mathcal{C}_j$, $r_j = \max_{i\in\mathcal{C}_j}|z_i|$, $r_B = \max_j r_j$, and $D_{K_c}(B) = \max_j \max_{i\in\mathcal{C}_j}\|\mathbf{k}_i - \mathbf{c}_j\|_2$,*

$$0 \le e_B^{\mathrm{SC}} \le \log\sum_j \pi_j e^{r_j^2/2} \le \frac{1}{2}r_B^2 \le \frac{1}{2}\|\mathbf{a}\|_2^2 D_{K_c}(B)^2, \tag{27}$$

where $\pi_j = n_j e^{\mathbf{a}^\top\mathbf{c}_j}/\widehat{M}_{\mathrm{SC}}$; a variance-sensitive refinement is given in the proof below.

Proof of Theorem 9. (A) Write $x_j = \mathbf{a}^\top\mathbf{c}_j$ and $\bar{x} = \sum_j w_j x_j = \mathbf{a}^\top\boldsymbol{\mu}_B$, so that $Y_j = x_j - \bar{x}$, $\sum_j w_j Y_j = 0$, and

$$\widehat{L}_{\mathrm{SC}} - \widehat{L}_{\mathrm{mean}} = \log\sum_j w_j e^{x_j} - \bar{x} = \log\sum_j w_j e^{Y_j}.$$

Applying Hoeffding's lemma (Hoeffding, 1963) to Y under the w-weighted distribution on $\{Y_j\}$ (mean zero, $|Y_j| \le R_{\mathrm{bet}}$) gives the upper bound $\sum_j w_j e^{Y_j} \le e^{R_{\mathrm{bet}}^2/2}$. For the lower bound, the Taylor integral remainder $e^y = 1 + y + y^2\int_0^1 (1-t)e^{ty}\,dt$ with $e^{ty} \ge e^{-R_{\mathrm{bet}}}$ for $|y| \le R_{\mathrm{bet}}$ gives $e^y \ge 1 + y + \frac{1}{2}e^{-R_{\mathrm{bet}}}y^2$; summing with weights w_j and using $\sum_j w_j Y_j = 0$,

$$\sum_j w_j e^{Y_j} \ge 1 + \frac{e^{-R_{\mathrm{bet}}}}{2}\sum_j w_j Y_j^2 = 1 + \frac{e^{-R_{\mathrm{bet}}}}{2}\mathbf{a}^\top\boldsymbol{\Sigma}_{\mathrm{between}}\mathbf{a},$$

and taking logarithms proves the lower bound.

(B) Since $z_i = \mathbf{a}^\top(\mathbf{k}_i - \mathbf{c}_j)$ has zero mean within $\mathcal{C}_j$ and $|z_i| \le r_j$,

$$\frac{M_B}{\widehat{M}_{\mathrm{SC}}} = \sum_{j=1}^{K_c}\pi_j\left[\frac{1}{n_j}\sum_{i\in\mathcal{C}_j} e^{z_i}\right] \le \sum_j \pi_j e^{r_j^2/2} \le e^{r_B^2/2},$$

using Hoeffding's lemma and $\sum_j \pi_j = 1$. Taking logarithms gives $e_B^{\mathrm{SC}} \le \log \sum_j \pi_j e^{r_j^2/2} \le \frac{1}{2} r_B^2$. For the variance-sensitive refinement stated in Theorem 9(B), let $\sigma_j^2 = \mathbf{a}^\top \mathbf{\Sigma}_j \mathbf{a}$ and $\phi(r) = (e^r - 1 - r)/r^2$ with $\phi(0) = \frac{1}{2}$. The same Taylor identity gives

$$e^z = 1 + z + z^2 \int_0^1 (1-t) e^{tz}\, dt, \qquad \int_0^1 (1-t) e^{tz}\, dt \le \int_0^1 (1-t) e^{t r_j}\, dt = \phi(r_j) \quad (z \le r_j),$$

so, using $\sum_{i \in \mathcal{C}_j} z_i = 0$ and $\frac{1}{n_j} \sum_{i \in \mathcal{C}_j} z_i^2 = \sigma_j^2$,

$$\frac{1}{n_j} \sum_{i \in \mathcal{C}_j} e^{z_i} \le 1 + \phi(r_j) \sigma_j^2, \qquad \frac{M_B}{\widehat{M}_{\mathrm{SC}}} \le 1 + \sum_j \pi_j \phi(r_j) \sigma_j^2,$$

which combines with the first bound as the minimum in the refinement. The final link $r_B \le \|\mathbf{a}\|_2 D_{K_c}(B)$ is Cauchy–Schwarz. □

Corollary 10 (Multiplicative coreset guarantee)**.** *Let $A > 0$ and $\varepsilon \ge 0$. For all queries with $\|\mathbf{a}\|_2 \le A$,*

$$e^{-\frac{1}{2} A^2 D_{K_c}(B)^2}\, M(\mathbf{q}, B) \;\le\; \widehat{M}_{\mathrm{SC}}(\mathbf{q}, B) \;\le\; M(\mathbf{q}, B),$$

so the radius budget $D_{K_c}(B) \le \sqrt{2\varepsilon}/A$ certifies a multiplicative $(e^{-\varepsilon}, 1)$ empirical coreset simultaneously for every such query.

Proof of Corollary 10. Exponentiate the chain of Theorem 9(B) with $\|\mathbf{a}\|_2 \le A$: $\widehat{M}_{\mathrm{SC}} \ge e^{-\frac{1}{2} A^2 D^2} M$. Setting $\frac{1}{2} A^2 D^2 \le \varepsilon$ gives the stated radius budget. □

Theorem 1 shows that the *exact* SubCenSA residual is pointwise no larger than the exact mean-centroid residual. This does not imply that the radius-based certificate $\frac{1}{2} r_B^2$ is no larger than the corresponding mean-centroid radius certificate: the maximum projected radius need not decrease under an arbitrary partition refinement.

The following proposition gives a distortion-sensitive complement to the covering-radius certificate.

Proposition 11 (SSE certificate)**.** *Fix a query direction $\mathbf{a}$ and a partition $\mathcal{P}$ of the block B into parts $\mathcal{C}_j$ with counts n_j and exact mean centroids $\mathbf{c}_j$. With the block mean $\boldsymbol{\mu}_B$, block radius $R_B = \max_{i \in B} \|\mathbf{k}_i - \boldsymbol{\mu}_B\|_2$, and distortion $W_B = \sum_j \sum_{i \in \mathcal{C}_j} \|\mathbf{k}_i - \mathbf{c}_j\|_2^2$,*

$$e_B^{\mathrm{SC}} \;\le\; U_W(\mathbf{a}, \mathcal{P}) := \log\Big(1 + \frac{e^{\|\mathbf{a}\|_2 R_B}}{2b}\, \|\mathbf{a}\|_2^2\, W_B\Big),$$

and on the right-hand side only W_B depends on the partition.

Proof of Proposition 11. Write $t_i = \mathbf{a}^\top \mathbf{k}_i$, $\bar{t}_j = \mathbf{a}^\top \mathbf{c}_j$, and $\bar{t} = \mathbf{a}^\top \boldsymbol{\mu}_B$, so that $z_i = t_i - \bar{t}_j$ has zero mean within $\mathcal{C}_j$ and, because the centroids are exact means, $\sum_j \frac{n_j}{b} \bar{t}_j = \bar{t}$. Two anchors carry the argument.

Anchor 1 (numerator cap). Since $\bar{t}_j$ is an average of $\{t_i\}_{i \in \mathcal{C}_j}$,

$$\bar{t}_j + \max_{i \in \mathcal{C}_j} z_i^+ = \max\Big\{\bar{t}_j,\ \max_{i \in \mathcal{C}_j} t_i\Big\} \le \max_{i \in B} t_i \le \bar{t} + \|\mathbf{a}\|_2 R_B,$$

the last step by Cauchy–Schwarz and the definition of R_B.

Anchor 2 (denominator floor). AM–GM applied to the convex-combination weights n_j/b gives

$$\widehat{M}_{\mathrm{SC}} = b \sum_j \frac{n_j}{b} e^{\bar{t}_j} \ge b \exp\Big(\sum_j \frac{n_j}{b} \bar{t}_j\Big) = b\, e^{\bar{t}}.$$

The proof rests on a one-sided Taylor bound: for real z with $z^+ = \max\{z, 0\}$, the integral form of the remainder, $e^z = 1 + z + z^2 \int_0^1 (1-t) e^{tz}\, dt$, has an integrand of at most e^{z^+} and $\int_0^1 (1-t)\, dt = \frac{1}{2}$,

so $e^z \le 1 + z + \frac{1}{2}z^2 e^{z^+}$. Combined with the vanishing within-cluster means of the z_i, this gives, for each part,

$$\sum_{i\in\mathcal{C}_j} e^{z_i} \le n_j + \frac{e^{\max_i z_i^+}}{2} \sum_{i\in\mathcal{C}_j} z_i^2.$$

Multiplying by $e^{\bar{t}_j}$, summing over j, and using $e^{\bar{t}_j}\sum_{i\in\mathcal{C}_j} e^{z_i} = M_B$ and $\sum_j n_j e^{\bar{t}_j} = \widehat{M}_{\mathrm{SC}}$,

$$M_B \le \widehat{M}_{\mathrm{SC}} + \frac{1}{2}\sum_j e^{\bar{t}_j + \max_i z_i^+} \sum_{i\in\mathcal{C}_j} z_i^2 \le \widehat{M}_{\mathrm{SC}} + \frac{1}{2} e^{\bar{t} + \|\mathbf{a}\|_2 R_B} \|\mathbf{a}\|_2^2 W_B,$$

where the second inequality applies Anchor 1 and Cauchy–Schwarz ($z_i^2 \le \|\mathbf{a}\|_2^2 \|\mathbf{k}_i - \mathbf{c}_j\|_2^2$). Dividing by $\widehat{M}_{\mathrm{SC}}$ and inserting the Anchor-2 floor,

$$\frac{M_B}{\widehat{M}_{\mathrm{SC}}} \le 1 + \frac{e^{\|\mathbf{a}\|_2 R_B}}{2b} \|\mathbf{a}\|_2^2 W_B,$$

and taking logarithms proves the claim. □

The quadratic leading constant is locally sharp. For $K_c = 1$ and two keys at $\boldsymbol{\mu}_B \pm R_B \mathbf{a}/\|\mathbf{a}\|_2$, the exact ratio is $\cosh t = 1 + \frac{1}{2}t^2 + O(t^4)$, while the exponentiated bound is $1 + \frac{1}{2}t^2 + O(t^3)$ for $t = \|\mathbf{a}\|_2 R_B$. Thus no smaller uniform quadratic coefficient is possible near $t = 0$; this does not claim global sharpness of the full exponential prefactor.

E.5 Ranking Guarantees: Proofs and Corollaries

Fix a query and a finite block set $\mathcal{B} = \{B_1, \ldots, B_N\}$. Let $\widehat{L}_B$ be any lower-bound log-mass estimator, define the one-sided residual $e_B = L_B - \widehat{L}_B \ge 0$, and use a fixed deterministic tie-breaking rule.

Theorem 12 (Ranking and mass guarantees under one-sided residuals)**.** *Fix a deterministic tie-breaking rule. Let $S_m^\star$ be the true top-m block set by L_B, and let $\widehat{S}_m$ be the top-m set selected by $\widehat{L}_B$. Define the boundary margin $\Delta_m = \min_{B\in S_m^\star,\, C\notin S_m^\star}(L_B - L_C)$, with $\Delta_m = 0$ if the boundary is tied.*

(i) *If $L_B > L_C$ and $e_B < L_B - L_C$, then $\widehat{L}_B > \widehat{L}_C$.*

(ii) *If $\max_{B\in S_m^\star} e_B < \Delta_m$, then $\widehat{S}_m = S_m^\star$.*

(iii) *If $\epsilon_\star = \max_{B\in S_m^\star} e_B$, then $\sum_{B\in\widehat{S}_m} M_B \ge e^{-\epsilon_\star}\sum_{B\in S_m^\star} M_B$.*

(iv) *The additive regret obeys $M(S_m^\star) - M(\widehat{S}_m) \le \sum_{B\in S_m^\star\setminus\widehat{S}_m}(M_B - \widehat{M}_B)$: the lost mass is bounded by the estimation gap on the missed true top-m blocks alone.*

Proof of Theorem 12. (i) Since $\widehat{L}_B = L_B - e_B$ and $e_C \ge 0$, the condition $e_B < L_B - L_C$ gives $\widehat{L}_B = L_B - e_B > L_C \ge \widehat{L}_C$.

(ii) If some $B \in S_m^\star$ were missed by $\widehat{S}_m$, then some $C \in \widehat{S}_m \setminus S_m^\star$ would be selected over it, so $L_C \ge \widehat{L}_C \ge \widehat{L}_B = L_B - e_B$ and hence $e_B \ge L_B - L_C \ge \Delta_m$. Thus $\max_{B\in S_m^\star} e_B < \Delta_m$ precludes missing any true top-m block, and $\widehat{S}_m = S_m^\star$.

(iii) Let $A = S_m^\star \setminus \widehat{S}_m$ and $F = \widehat{S}_m \setminus S_m^\star$, so $|A| = |F|$. For any bijection $\phi : A \to F$, the selection property gives, for each $C \in A$,

$$L_C = \widehat{L}_C + e_C \le \widehat{L}_{\phi(C)} + \epsilon_\star \le L_{\phi(C)} + \epsilon_\star,$$

so $M_C \le e^{\epsilon_\star} M_{\phi(C)}$ and $M(A) \le e^{\epsilon_\star} M(F)$. Therefore $M(S_m^\star) = M(S_m^\star \cap \widehat{S}_m) + M(A) \le e^{\epsilon_\star}[M(S_m^\star \cap \widehat{S}_m) + M(F)] = e^{\epsilon_\star} M(\widehat{S}_m)$, and rearranging gives the selected-mass guarantee.

(iv) With $U = S_m^\star \setminus \widehat{S}_m$ and $F = \widehat{S}_m \setminus S_m^\star$, the top-$m$ property of $\widehat{L}$ gives $\widehat{M}(F) \ge \widehat{M}(U)$ (F consists of $|F|$ of the m largest estimated masses, U of none), and one-sidedness gives $M(F) \ge$

$\widehat{M}(F)$. Hence $M(S_m^\star) - M(\widehat{S}_m) = M(U) - M(F) \le M(U) - \widehat{M}(U) = \sum_{B \in S_m^\star \setminus \widehat{S}_m} (M_B - \widehat{M}_B)$. □

The corollaries below instantiate Theorem 12 for SubCenSA through the per-block certificate η_B^{SC}, the pointwise minimum of the residual certificates of Theorem 9(B) and the SSE certificate U_W.

Corollary 13 (SubCenSA top-m certificates)**.** *For SubCenSA define the per-block certificate*

$$\eta_B^{\mathrm{SC}} := \min\Big\{\log \sum_j \pi_j e^{r_j^2/2},\ U_W(\mathbf{a}, \mathcal{P}),\ \tfrac{1}{2} r_B^2,\ \tfrac{1}{2}\|\mathbf{a}\|_2^2 D_{K_c}(B)^2\Big\},$$

the pointwise minimum of the residual certificates of Theorem 9(B) *and the SSE certificate U_W of Proposition 11, and let $\eta_\star^{\mathrm{SC}} = \max_{B \in S_m^\star} \eta_B^{\mathrm{SC}}$.* (i) *If $\eta_\star^{\mathrm{SC}} < \Delta_m$, then $\widehat{S}_m^{\mathrm{SC}} = S_m^\star$; a sufficient geometric condition is $\frac{1}{2}\|\mathbf{a}\|_2^2 \max_{B \in S_m^\star} D_{K_c}(B)^2 < \Delta_m$.* (ii) $\sum_{B \in \widehat{S}_m^{\mathrm{SC}}} M_B \ge e^{-\eta_\star^{\mathrm{SC}}} \sum_{B \in S_m^\star} M_B \ge \exp\big(-\frac{1}{2}\|\mathbf{a}\|_2^2 \max_{B \in S_m^\star} D_{K_c}(B)^2\big) \sum_{B \in S_m^\star} M_B$.

Proof. Each term of the minimum upper-bounds the one-sided residual—the first by Theorem 9(B), U_W by Proposition 11, and the remaining two by domination within Eq. equation 27—so $e_B^{\mathrm{SC}} \le \eta_B^{\mathrm{SC}}$ with the chain compatibility $e_B^{\mathrm{SC}} \le \eta_B^{\mathrm{SC}} \le \frac{1}{2} r_B^2 \le \frac{1}{2}\|\mathbf{a}\|_2^2 D_{K_c}(B)^2$. Both claims are then Theorem 12(ii)–(iii) with $\epsilon_\star = \eta_\star^{\mathrm{SC}} \le \frac{1}{2}\|\mathbf{a}\|_2^2 \max_{B \in S_m^\star} D_{K_c}(B)^2$. □

The minimum in η_B^{SC} cannot be dispensed with: the SSE certificate alone need not lie below $\frac{1}{2} r_B^2$, because W_B is a full-dimension distortion whereas r_B is a projected radius—keys spread in directions orthogonal to $\mathbf{a}$ can make r_B vanish while W_B stays positive.

Kendall and Spearman certificates. Assume no exact true-rank ties, let $\mathrm{Inv}(\widehat{L}, L)$ denote the number of block pairs with $L_B > L_C$ but $\widehat{L}_B \le \widehat{L}_C$, and suppose $e_B \le \epsilon$ for all B. Theorem 12(i) gives

$$\mathrm{Inv}(\widehat{L}, L) \le \mathcal{M}_\epsilon := \#\{(B, C) : L_B > L_C,\ L_B - L_C \le \epsilon\}, \qquad \tau(\widehat{L}, L) \ge 1 - \frac{2\mathcal{M}_\epsilon}{\binom{N}{2}}.$$

Writing $C_\epsilon = \max_B \#\{C \ne B : |L_B - L_C| \le \epsilon\}$ for the largest ϵ-margin neighborhood of any block, each block's rank displacement is at most C_ϵ, and therefore $\rho_{\mathrm{Spearman}} \ge 1 - 6C_\epsilon^2/(N^2 - 1)$. These are margin-based certificates; they do not claim unconditional ranking dominance over arbitrary indices.

Theorem 14 (Computable normalized-probability interval certificate)**.** *Fix a head and its atoms. Suppose each atom x is known to satisfy*

$$\widehat{M}_x \le M_x \le \widehat{M}_x e^{U_x}$$

for a computable bound $U_x \ge 0$. Then the normalized probability admits the interval

$$p_x \ \ge\ \frac{\widehat{M}_x}{\widehat{M}_x + \sum_{y \ne x} \widehat{M}_y e^{U_y}}, \qquad p_x \ \le\ \frac{\widehat{M}_x e^{U_x}}{\widehat{M}_x e^{U_x} + \sum_{y \ne x} \widehat{M}_y}.$$

Writing $\ell_{h,x}, u_{h,x}$ for these per-head bounds and $\ell_G(x) = \sum_{h \in G} \beta_h \ell_{h,x}$, $u_G(x) = \sum_{h \in G} \beta_h u_{h,x}$ for the group bounds, the realized selection $\widehat{S}_m$ is certified to be the true normalized group top-m whenever

$$\min_{B \in \widehat{S}_m} \ell_G(B) \ >\ \max_{C \in \mathcal{O} \setminus \widehat{S}_m} u_G(C).$$

Proof of Theorem 14. $p_x = M_x/(M_x + \sum_{y \ne x} M_y)$ is increasing in M_x and decreasing in every other mass, so substituting the least favorable endpoints gives the two bounds; under the final condition every selected block's true group score exceeds every outsider's, so the selection is the true top-m. □

The interval certificate is an offline diagnostic: computing it requires the bounds U_x (from Theorem 9 or the SSE certificate U_W) but no change to the deployed selector, and it introduces no dynamic budget.

Proposition 15 (Pointwise tightening does not imply uniformly better ranking)**.** *There exist two lower-bound estimators $\widehat{L}^{(1)}$ and $\widehat{L}^{(2)}$ such that*

$$\widehat{L}_B^{(1)} \le \widehat{L}_B^{(2)} \le L_B \quad \textit{for every block } B,$$

but $\widehat{L}^{(1)}$ ranks a pair correctly while $\widehat{L}^{(2)}$ ranks it incorrectly.

Proof. Take two blocks with $L_{B_1} = 1.00$ and $L_{B_2} = 0.95$. Define $\widehat{L}^{(1)}_{B_1} = 0.80$, $\widehat{L}^{(1)}_{B_2} = 0.65$, and $\widehat{L}^{(2)}_{B_1} = 0.80$, $\widehat{L}^{(2)}_{B_2} = 0.95$. Both estimators are lower bounds, and $\widehat{L}^{(1)}_B \le \widehat{L}^{(2)}_B \le L_B$ for both blocks. Estimator 1 ranks B_1 above B_2, correctly. Estimator 2 ranks B_2 above B_1, incorrectly. □

E.6 Self-Normalized GQA Selection

Theorem 16 (Self-normalization error is controlled by residual oscillation)**.** *Fix a head and view the mandatory set and each optional block as atoms with positive mass estimates $\widehat{M}_x > 0$ and log-residuals $e_x = \log M_x - \log \widehat{M}_x$ (of either sign, for a general estimator). Let $\widehat{Z} = \sum_x \widehat{M}_x$, $\widehat{p}_x = \widehat{M}_x/\widehat{Z}$, and let the oscillation be $\omega = \max_x e_x - \min_x e_x$. Then*

$$e^{-\omega} p_x \;\le\; \widehat{p}_x \;\le\; e^{\omega} p_x \qquad \textit{and} \qquad \mathrm{TV}(p, \widehat{p}) \le \tanh(\omega/4).$$

Proof of Theorem 16. (Multiplicative bound.) With $\lambda = -\log \sum_x p_x e^{-e_x}$, we have $\widehat{Z} = Z \sum_x p_x e^{-e_x}$ and $\log \frac{\widehat{p}_x}{p_x} = \log \frac{\widehat{M}_x}{\widehat{Z}} - \log \frac{M_x}{Z} = -e_x + \lambda$ for every atom. Since $\sum_x p_x e^{-e_x}$ is a p-weighted average of values in $[e^{-\max_x e_x}, e^{-\min_x e_x}]$, we have $\min_x e_x \le \lambda \le \max_x e_x$, and therefore $|\lambda - e_x| \le \omega$ for every x; exponentiating gives $e^{-\omega} p_x \le \widehat{p}_x \le e^{\omega} p_x$.

(TV bound.) The sharp bound $\mathrm{TV} \le \tanh(\omega/4)$ for a likelihood ratio of range e^{ω} is the classical-probability special case of the Hilbert-metric inequality $\frac{1}{2}\|b_1 - b_2\|_B \le \tanh\big(h_C(b_1, b_2)/4\big)$ of Reeb et al. (2011) (their Proposition 7; the weaker classical range bound $\mathrm{TV} \le (M-m)/(M+m) = \tanh(\omega/2)$ is its one-line consequence); we give a short self-contained derivation. Consider $Y = e^{-e_x}$ under the true atom distribution p_x, taking values in $[m, M]$ with $M/m = e^{\omega}$ and mean μ. Since $\widehat{p}_x = p_x Y_x/\mu$,

$$\mathrm{TV}(p, \widehat{p}) = \frac{1}{2} \sum_x p_x \left| 1 - \frac{Y_x}{\mu} \right| = \frac{\mathbb{E}|Y - \mu|}{2\mu}.$$

Among all distributions on $[m, M]$ with mean μ, $\mathbb{E}|Y-\mu|$ is maximized by the two-point distribution on $\{m, M\}$ with $\mathbb{P}(Y{=}M) = (\mu - m)/(M - m)$ (interior mass can be pushed to the endpoints without changing the mean while increasing the mean absolute deviation from μ, by convexity of $|y - \mu|$), for which $\mathbb{E}|Y - \mu| = 2(M - \mu)(\mu - m)/(M - m)$. Hence $\mathrm{TV}(p, \widehat{p}) \le (\mu - m)(M - \mu)/[\mu(M - m)]$, a concave function of μ whose maximum is attained at $\mu = \sqrt{Mm}$ and equals $(\sqrt{M} - \sqrt{m})^2/(M - m) = (\sqrt{M} - \sqrt{m})/(\sqrt{M} + \sqrt{m}) = \tanh(\omega/4)$, using $\sqrt{M/m} = e^{\omega/2}$. □

Theorem 17 (Guarantees of the default normalized GQA selector)**.** *Let $J(B) = \sum_{h \in G} \beta_h p_h(B)$ and $\widehat{J}(B) = \sum_{h \in G} \beta_h \widehat{p}_h(B)$ with $\beta_h \ge 0$, $\sum_{h \in G} \beta_h = 1$; let $\Omega = \max_{h \in G} \omega_h$ and $\tau_G = \sum_{h \in G} \beta_h \tanh(\omega_h/4)$. Then*

$$e^{-\Omega} J(B) \le \widehat{J}(B) \le e^{\Omega} J(B), \qquad \mathrm{TV}(J, \widehat{J}) \le \tau_G.$$

Moreover, with $S^\star_m$ the true top-m by J over the optional domain and $\widehat{S}_m$ the top-m by $\widehat{J}$, both under one deterministic tie-breaking rule:

(A) *if $\min_{B \in S^\star_m,\, C \notin S^\star_m} [\log J(B) - \log J(C)] > 2\Omega$, then $\widehat{S}_m = S^\star_m$;*

(B) *$J(\widehat{S}_m) \ge e^{-2\Omega} J(S^\star_m)$;*

(C) $J(S_m^\star) - J(\widehat{S}_m) \le 2\tau_G$;

(D) *the attention-output closure of Corollary 6 holds with the selected set:*

$$\sum_{h\in G} \beta_h \|\mathbf{o}_h - \mathbf{o}_{h,\mathcal{A}\cup\widehat{S}_m}\|_2 \le 2V\Big[1 - J(\mathcal{A}) - e^{-2\Omega} J(S_m^\star)\Big], \tag{28}$$

and equivalently $\sum_h \beta_h \|\mathbf{o}_h - \mathbf{o}_{h,\mathcal{A}\cup\widehat{S}_m}\|_2 \le 2V\big[1 - J(\mathcal{A}) - \max\{0,\, J(S_m^\star) - 2\tau_G\}\big]$.

Proof of Theorem 17. (Group bounds.) β_h-weighting the per-head display of Theorem 16 gives $e^{-\Omega} J(B) \le \widehat{J}(B) \le e^{\Omega} J(B)$, and convexity of TV in each argument (equivalently, the coupling representation $\mathrm{TV}(\mu,\nu) = \inf \mathbb{P}[X \ne Y]$) gives $\mathrm{TV}(J,\widehat{J}) \le \sum_h \beta_h \,\mathrm{TV}(p_h, \widehat{p}_h) \le \sum_h \beta_h \tanh(\omega_h/4) = \tau_G$.

(A) By the group bound, $|\log \widehat{J}(B) - \log J(B)| \le \Omega$ for every block; a true log-margin exceeding 2Ω between any $B \in S_m^\star$ and any outsider C then forces $\log \widehat{J}(B) > \log \widehat{J}(C)$ for all such pairs, so the estimated top-m equals $S_m^\star$.

(B) Using the group bound twice and the top-m property of $\widehat{J}$,

$$J(\widehat{S}_m) \ge e^{-\Omega} \widehat{J}(\widehat{S}_m) \ge e^{-\Omega} \widehat{J}(S_m^\star) \ge e^{-2\Omega} J(S_m^\star).$$

(C) For any atom set S, $|J(S) - \widehat{J}(S)| \le \mathrm{TV}(J,\widehat{J})$, so $J(S_m^\star) - J(\widehat{S}_m) \le [J(S_m^\star) - \widehat{J}(S_m^\star)] + [\widehat{J}(\widehat{S}_m) - J(\widehat{S}_m)] \le 2\tau_G$.

(D) Apply Corollary 6 with $S = \widehat{S}_m$ and insert (B):

$$\sum_h \beta_h \|\mathbf{o}_h - \mathbf{o}_{h,\mathcal{A}\cup\widehat{S}_m}\|_2 \le 2V\big[1 - J(\mathcal{A}) - J(\widehat{S}_m)\big] \le 2V\big[1 - J(\mathcal{A}) - e^{-2\Omega} J(S_m^\star)\big],$$

and the additive form follows from $J(\widehat{S}_m) \ge J(S_m^\star) - 2\tau_G$ (by (C)) together with $J(\widehat{S}_m) \ge 0$. □

Corollary 18 (Oracle-free output closure from the realized read)**.** *Equation equation 28 is oracle-relative: it is stated in terms of $J(S_m^\star)$ and Ω. A counterpart free of oracle quantities follows from the realized read alone. Fix a head h and let $R = \mathcal{A} \cup \widehat{S}_m$ be the realized read set, whose exact normalizer contribution $Z_{R,h} = \sum_{i\in R} e^{s_{h,i}}$ is computed by the realized attention. Suppose every unselected optional block carries a certificate $M_h(B) \le \widehat{M}_h(B) e^{U_h(B)}$ (any of the upper certificates of Sections 3.1 and 3.2, inflated by the precision terms of Theorem 22 for the implemented scores). Then*

$$\alpha_h(R) \;\ge\; \frac{Z_{R,h}}{Z_{R,h} + \sum_{B\in\mathcal{O}\setminus\widehat{S}_m} \widehat{M}_h(B)\, e^{U_h(B)}},$$

and substituting this lower bound on $\alpha_h(R)$ into Eq. equation 25b bounds the per-head output error (and, after β_h-weighting, the group bound of Corollary 6) using only the realized normalizer, the stored estimates, and the certificates.

Proof of Corollary 18. Since $Z_h = Z_{R,h} + \sum_{B\notin R} M_h(B)$ and the mandatory mass is part of $Z_{R,h}$, the certificates on the unselected optional blocks give $Z_h \le Z_{R,h} + \sum_{B\in\mathcal{O}\setminus\widehat{S}_m} \widehat{M}_h(B) e^{U_h(B)}$, so the displayed ratio lower-bounds $\alpha_h(R) = Z_{R,h}/Z_h$; substituting it for $\alpha_h(R)$ in Eq. equation 25b gives the claim. □

Substituting the construction certificates. Substituting the certificates $0 \le e_h(B) \le U_h(B)$ (Theorem 9(B) or U_W) into the two theorems above yields $\omega_h \le \max_B U_h(B)$ and $\Omega \le \max_h \max_B U_h(B)$—the certificate chain instantiated in Section 3.2.

F Necessity of Count Weighting

This section complements the proofs in Appendix E with the necessity of the count weights in the mass-dominance and ranking behavior.

F.1 Necessity of the Count Weights

Define the unweighted sub-centroid score $U(\mathbf{q}, B) = \sum_{j=1}^{K_c} e^{\mathbf{a}^\top \mathbf{c}_j}$. Because $n_j \geq 1$,

$$U(\mathbf{q}, B) \leq \sum_j n_j e^{\mathbf{a}^\top \mathbf{c}_j} = \widehat{M}_{\mathrm{SC}}(\mathbf{q}, B) \leq M(\mathbf{q}, B),$$

so U is still a lower bound on the true mass. However, it does not satisfy the key improvement over mean pooling.

Proposition 19 (Unweighted prototype scores do not preserve mean dominance)**.** *There exist blocks and partitions for which $U(\mathbf{q}, B) < \widehat{M}_{\mathrm{mean}}(\mathbf{q}, B)$; therefore the left inequality $\widehat{M}_{\mathrm{mean}} \leq \widehat{M}_{\mathrm{SC}}$ of Theorem 1 is not preserved by the unweighted variant.*

Proof. Take $K_c = 2$, $n_1 = n_2 = b/2$, and suppose $\mathbf{a}^\top \mathbf{c}_1 = \mathbf{a}^\top \mathbf{c}_2 = x$. Then $U = 2e^x$, while the block mean also has projected logit x, so $\widehat{M}_{\mathrm{mean}} = b\, e^x$. For any $b > 2$, $U = 2e^x < b\, e^x = \widehat{M}_{\mathrm{mean}}$. □

Beyond losing the mean-dominance chain, the unweighted score can also invert the true mass ordering, so the counts are not a calibration constant that could be repaired a posteriori.

Proposition 20 (Unweighted prototype scores can invert the true ranking)**.** *There exist two blocks with equal size b and partitions with constant in-cluster logits such that $M(\mathbf{q}, B_2) > M(\mathbf{q}, B_1)$ while $U(\mathbf{q}, B_1) > U(\mathbf{q}, B_2)$; moreover, the inversion persists under any common rescaling of U (in particular the uniform weights b/K_c).*

Proof. Let $K_c = 2$ and $b = 64$. In block B_1, one key has query-projected logit 2 and the other 63 have logit 0, with the two subclusters separating them; in block B_2, 32 keys have logit 1 and 32 have logit 0. The count-weighted estimates are exact:

$$M(\mathbf{q}, B_1) = \widehat{M}_{\mathrm{SC}}(B_1) = e^2 + 63 \approx 70.4, \qquad M(\mathbf{q}, B_2) = \widehat{M}_{\mathrm{SC}}(B_2) = 32e + 32 \approx 119.0,$$

so the true mass of B_2 is larger. The unweighted scores, however, satisfy

$$U(B_1) = e^2 + 1 \approx 8.4 \qquad \text{vs.} \qquad U(B_2) = e + 1 \approx 3.7,$$

which ranks B_1 above B_2; the error is an omitted multiplicity, not a common offset, so multiplying U by any constant (such as $b/K_c = 32$) leaves the inversion unchanged. □

G Design-Choice Analysis

This section analyzes the design choices behind the SubCenSA index: the self-normalized grouped-query aggregation (with the mass-space sum retained as an ablation), the uniqueness of the temperature $\tau = 1$, and robustness to finite-precision score perturbations.

G.1 Grouped-Query Attention Aggregation

The default aggregation of Eq. equation 13—the self-normalized probability sum with uniform weights—is chosen because it is the estimator of $J_G(B)$, the average head-level probability retention that appears in the output-error proxy of Corollary 6 and in the selection guarantees of Theorem 17. It also restores each head's invariance to per-head logit shifts, which raw masses violate: adding a constant to all logits of one head rescales every raw $M_h(B)$ by a head-specific factor and can reorder the raw-sum scores even though every head's softmax is unchanged. The mass-space sum, the raw and normalized maxima, and per-head selection are therefore evaluated only as ablations; their consistency properties are recorded here for completeness.

For completeness we record the consistency of the raw mass-space sum used in the ablation. If each head satisfies $L_h(B) - \widehat{L}_{\mathrm{SC},h}(B) \leq \varepsilon_h(B)$, then for fixed nonnegative weights γ_h the sums $\widehat{M}_G(B) = \sum_{h \in G} \gamma_h \widehat{M}_{\mathrm{SC},h}(B)$ and $M_G(B) = \sum_{h \in G} \gamma_h M_h(B)$ obey the sandwich

$\widehat{M}_G(B) \le M_G(B)$ and $\log M_G(B) - \log \widehat{M}_G(B) \le \max_{h\in G} \varepsilon_h(B)$: summing the per-head estimates $e^{-\varepsilon_h(B)} M_h(B) \le \widehat{M}_{\mathrm{SC},h}(B) \le M_h(B)$ with weights γ_h gives the first display, and taking logarithms the second. The ranking guarantees of Theorem 12 therefore transfer to the raw-sum objective M_G.

This consistency statement concerns the additive raw objective, not one aligned with the output-error proxy. The *raw* maximum $\max_h M_h(B)$ shares this logit-shift sensitivity and tracks a single head's raw response; the *normalized* maximum $\max_h p_h(B)$ is shift-invariant, like the default selector, but optimizes a different objective—the best single head's retention rather than the average—so neither is a drop-in replacement for the default. Per-head selection leaves the shared index vectors readable once per group but multiplies the per-head logical compute and, since the heads' top-m sets generally differ, the selected-KV read traffic. None of these variants appears in the default method definition.

G.2 Temperature Family and Uniqueness of $\tau = 1$

Proposition 21 (Temperature limits and uniqueness of $\tau = 1$)**.** *Write the raw temperature score of Eq. equation 22 as $S_\tau(B) = \tau \log \sum_{j=1}^{K_c} n_j \exp(x_j/\tau)$ with $x_j = \mathbf{a}^\top \mathbf{c}_j$, and, for equal block sizes b, the block-shifted statistic*

$$R_\tau(B) = \log b + \tau \log \sum_{j=1}^{K_c} w_j \exp\left(\frac{x_j}{\tau}\right), \qquad w_j = \frac{n_j}{b},$$

which differs from S_τ by the block-independent offset $(1-\tau)\log b$. For $0 < \tau \le 1$,

$$S_\tau(B) \le S_1(B) = \widehat{L}_{\mathrm{SC}}(\mathbf{q}, B) \le L(\mathbf{q}, B),$$

so the raw score remains a one-sided lower score, with a residual no smaller than that of $\tau = 1$. The map $\tau \mapsto R_\tau(B)$ is continuous and non-increasing; it is strictly decreasing unless all x_j are equal. Moreover,

$$\lim_{\tau\to 0^+} R_\tau(B) = \log b + \max_j x_j, \qquad R_1(B) = \widehat{L}_{\mathrm{SC}}(\mathbf{q}, B), \qquad \lim_{\tau\to\infty} R_\tau(B) = \widehat{L}_{\mathrm{mean}}(\mathbf{q}, B).$$

Thus $\tau = 1$ is the unique temperature whose raw score is identically the logarithm of the count-weighted coreset mass $\widehat{M}_{\mathrm{SC}}$ for every block with $b > 1$. For $\tau > 1$, the normalized score R_τ is a looser lower score approaching mean pooling. For $\tau < 1$, both scores move toward a hard centroid maximum: the raw score S_τ stays below $L(\mathbf{q}, B)$, whereas the shifted statistic R_τ no longer carries a lower-bound mass interpretation and may overestimate the dense block mass.

Proof sketch. Replicating each x_j exactly n_j times, e^{S_τ} is the ℓ_p norm of the b values e^{x_i} with $p = 1/\tau \ge 1$; norm monotonicity gives $S_\tau \le \widehat{L}_{\mathrm{SC}} \le L$, and uniqueness at $\tau = 1$ is immediate when all x_j coincide ($S_\tau = x + \tau \log b$). For monotonicity, write $u = 1/\tau$, $A(u) = \log \sum_j w_j e^{u x_j}$, so $R_\tau = \log b + A(u)/u$ with $(A(u)/u)' = (uA' - A)/u^2$ and $(A - uA')' = -uA'' \le 0$ since $A'' = \mathrm{Var}_{p_u}(x) \ge 0$; hence R_τ is non-increasing in τ, strictly unless all x_j coincide. The three limits follow from log-sum-exp asymptotics ($A(u) = u \max_j x_j + O(1)$ as $u \to \infty$, $A(u) = u \sum_j w_j x_j + o(u)$ as $u \to 0$), and $R_1 = \log \sum_j n_j e^{x_j} = \widehat{L}_{\mathrm{SC}}$. □

G.3 Finite-Precision Score Perturbations

The exact statements assume that stored centroids are exact empirical means and that the score is evaluated in exact arithmetic. The following theorem bounds the effect of the implemented score's quantization and reduction errors on each layer of the theory chain.

Theorem 22 (Finite-precision score perturbations)**.** (A) Single-head coreset score. *Let $u_j = \mathbf{a}^\top \mathbf{c}_j + \log n_j$ be the exact prototype log-scores, and let $\widetilde{u}_j$ be the implemented ones, including centroid quantization, query cast, dot-product accumulation, and log-count lookup error. Suppose $\max_j |\widetilde{u}_j - u_j| \le \eta_{\mathrm{proto}}$, and suppose the final log-sum-exp reduction adds at most η_{red}. With $\zeta = \eta_{\mathrm{proto}} + \eta_{\mathrm{red}}$ and $\widetilde{L}_{\mathrm{SC}} = \mathrm{logsumexp}_j \widetilde{u}_j$,*

$$\left|\widetilde{L}_{\mathrm{SC}} - \widehat{L}_{\mathrm{SC}}\right| \le \zeta, \qquad \widetilde{L}_{\mathrm{SC}} - \zeta \;\le\; L_B :$$

the deflated score is a certified lower bound on L_B, with log-residual at most $e_B^{\mathrm{SC}} + 2\zeta$.

(B) One-sided ordering. *If each implemented block score satisfies $|\widetilde{L}_B - \widehat{L}_B| \le \zeta_B$, then the pairwise ordering condition of Theorem 12*(i) *tightens to*

$$L_B - L_C > e_B + \zeta_B + \zeta_C,$$

which suffices for the implemented score to rank B above C.

(C) Normalized GQA selection. *Suppose each head's atom log-mass implementation error is at most ζ_h, and the normalizer and group-aggregation stages contribute at most δ to the final group log-scores. With*

$$\kappa = \max_{h \in G} \big(\omega_h + 2\zeta_h\big) + \delta,$$

the multiplicative selection guarantee of Theorem 17 persists with κ in place of Ω:

$$J(\widetilde{S}_m) \;\ge\; e^{-2\kappa} J(S_m^\star),$$

where $\widetilde{S}_m$ is the top-m set selected by the implemented group scores.

Proof of Theorem 22. (A) log-sum-exp is 1-Lipschitz in ℓ_∞, so $|\widetilde{L}_{\mathrm{SC}} - \widehat{L}_{\mathrm{SC}}| \le \zeta$; with $\widehat{L}_{\mathrm{SC}} \le L_B$ (Theorem 1), the deflated score satisfies $\widetilde{L}_{\mathrm{SC}} - \zeta \le L_B$ with residual at most $e_B^{\mathrm{SC}} + 2\zeta$. (B) $\widetilde{L}_B \ge L_B - e_B - \zeta_B > L_C + \zeta_C \ge \widetilde{L}_C$ under the displayed condition. (C) The implemented per-atom log-residual is $e_{h,x} - \xi_{h,x}$ with $|\xi_{h,x}| \le \zeta_h$, so the realized oscillation of head h is at most $\omega_h + 2\zeta_h$; repeating the chain of Theorem 17 with κ in place of Ω gives $J(\widetilde{S}_m) \ge e^{-2\kappa} J(S_m^\star)$. □

Claim (C), like Eq. equation 28, is oracle-relative: it is a guarantee stated in terms of the true top-m score, not an online computation. Its computable counterpart is Corollary 18, with the certificates $U_h(B)$ inflated by $2\zeta_h + \delta$ to absorb the implemented-score errors.

Implementation boundaries. The theorem states what a bounded implementation error does to the guarantees; it does not assert that the deployed kernel realizes those bounds. We do not require the deployed path to deflate its scores, and we do not present maximum errors measured on a finite test set as certified all-input error bounds. Any numerical validation must account for every listed source—construction rounding, the query cast, centroid quantization, dot-product accumulation, the log-count lookup, the normalizer reduction, and the group reduction—before the constants ζ, ζ_h, δ can be instantiated. Without an explicit perturbation correction, the finite-precision kernel score may break the exact lower-bound property.