ATTEND AT THE RIGHT RESOLUTION: HEAD-ADAPTIVE SPARSE ATTENTION VIA QUERY–KEY INTERACTION STRUCTURE

Anonymous authors
Paper under double-blind review

ABSTRACT

Attention is a major scalability bottleneck in large language models (LLMs), with its computational cost and key-value (KV) memory traffic becoming increasingly prohibitive as context length grows. Sparse attention addresses this bottleneck by restricting each query to a subset of the context, but its effectiveness hinges on identifying relevant regions both cheaply and reliably. Existing training-free methods primarily rely on runtime relevance estimates, with selection dimensionality and granularity typically governed by fixed heuristics rather than the intrinsic interaction structure of pretrained attention heads. We take a different perspective: the query–key operator of each pretrained head defines a characteristic response structure that determines the resolution needed for sparse selection, both across feature directions and across context tokens. Building on this insight, we propose Head-Adaptive Sparse Attention (HASA), a training-free sparse attention framework that jointly adapts selection resolution along these two dimensions. Along the feature dimension, the singular spectrum of each head's query–key operator determines an effective rank, yielding a compact, head-specific response space. Along the sequence dimension, neighboring regions are progressively merged according to the directional coherence of their token responses in this same space, while heterogeneous regions remain separate. Exact attention is then computed using the original full-dimensional KV states of the selected regions. HASA captures **88.1%** of dense-attention mass and, compared with UNIQUE, a strong recent sparse-attention baseline, improves accuracy by up to 3.47 and 2.57 points on RULER and LongBench-Pro, respectively, and accelerates self-attention by up to **1.32**×, while achieving up to **14.07**× speedup over dense attention.

1 INTRODUCTION

Large language models (LLMs) have become a general-purpose foundation for a wide range of applications, with attention serving as the core mechanism for integrating information across the input context. However, dense attention requires each query to interact with the entire available context, incurring substantial computational cost and key-value (KV) memory traffic as sequence lengths and inference workloads increase (OpenAI et al., 2024; Grattafiori et al., 2024; Comanici et al., 2025; Pope et al., 2023; Dao, 2024; Kwon et al., 2023). These growing costs make efficient attention increasingly important for scaling LLM inference to longer contexts and larger workloads.

Sparse attention reduces these costs by exploiting the observation that attention is often highly concentrated, allowing computation and KV access to be restricted to a small subset of the context while preserving most relevant interactions. Existing methods broadly follow two strategies. Approaches without per-query selection impose predefined access patterns or reduce the retained KV context using structural or historical signals, offering regular and predictable computation but limited ability to adapt to the interactions required by each query (Beltagy et al., 2020; Xiao et al., 2024b; 2025; Zhang et al., 2023; Li et al., 2024). Content-adaptive approaches, in contrast, estimate query-dependent relevance from runtime information to select relevant tokens, enabling sparse patterns to better capture input-dependent and long-range dependencies, albeit at the cost of additional selection computation (Yuan et al., 2025; Zhao et al., 2026). Despite these differences, both strategies face a fundamental challenge in determining the appropriate resolution for context selection. As illustrated in Fig. 1,

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.

coarse-grained selection groups relevant and irrelevant tokens into the same region, forcing a choice between accessing unnecessary KV states and missing important interactions. Fine-grained selection localizes relevant tokens more precisely, but requires evaluating and comparing more candidate regions across the context, even where neighboring tokens need not be distinguished. Moreover, attention heads differ in the token relationships they capture, so a selection granularity appropriate for one head may be too coarse or unnecessarily fine for another (Correia et al., 2019).

However, characterizing each head's full input-dependent sensitivity would be costly during inference. Instead, we leverage the distinct, input-independent interaction structure defined by each head's pretrained query–key transformations. This structure identifies the directions to which each head is most responsive, providing a basis for determining both important feature directions and consequential distinctions between context tokens. Along the feature dimension, strongly responsive directions carry the distinctions most consequential to relevance estimation, whereas weak-response directions contribute less and can be represented more coarsely during selection. Along the sequence dimension, the same response structure determines whether tokens need to be distinguished by that head: tokens that differ substantially in the original representation space may induce similar responses if their differences lie primarily along weak-response directions, whereas seemingly similar tokens may need to remain distinct when their differences align with strongly responsive directions. Token redundancy is therefore relative to the attention head rather than solely to the original token representations. Together, these observations motivate jointly adapting feature and sequence resolution according to the same head-specific interaction structure.

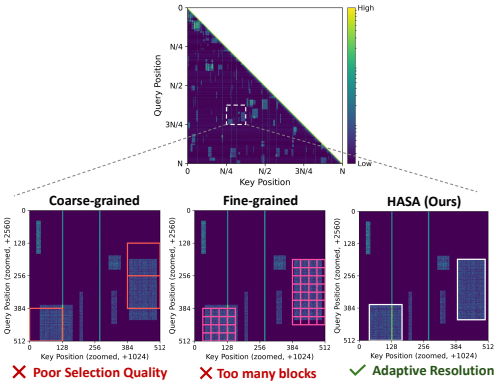


Figure 1: **Motivation for HASA.** Coarse-grained selection risks missing relevant tokens, while fine-grained selection increases selection overhead. HASA adapts selection granularity for each head.

Based on this insight, we propose **HASA**, a training-free sparse attention framework that adapts the resolution used for sparse selection along both feature and sequence dimensions. Along the feature dimension, HASA uses the singular spectrum of each pretrained query–key operator to determine a head-specific effective rank, retaining the dominant interaction directions needed for relevance estimation. Along the sequence dimension, tokens are represented in this compact head-specific space, and neighboring regions are progressively merged when their representations remain directionally coherent, while heterogeneous regions retain finer granularity. These compact representations are used only for routing; exact attention is computed over the original full-dimensional KV states of the selected regions. By reducing both the dimensionality of relevance estimation and the number of candidate regions, HASA lowers selection overhead while retaining head-relevant distinctions for sparse selection. The spectral structure and effective ranks are determined offline, avoiding input-dependent rank-selection overhead during inference.

Our contributions are as follows:

- We reveal that each head's pretrained query–key interaction structure characterizes its selection requirements along both feature and sequence dimensions, providing a unified basis for head-adaptive sparse attention.
- We develop a head-adaptive feature resolution mechanism that determines each head's effective rank using spectral entropy and a truncation-error criterion, yielding a compact response space that preserves dominant interaction directions.
- We develop a head-adaptive sequence resolution mechanism that progressively merges neighboring regions based on token-response coherence in the head-specific response space, yielding candidate regions of varying granularity.
- Extensive experiments show that HASA captures **88.1%** of dense-attention mass, improves accuracy over UNIQUE by up to **3.47** and **2.57** points on RULER and LongBench-Pro, respectively, and achieves self-attention speedups of up to **14.07**$\times$ over dense attention and **1.32**$\times$ over UNIQUE.

2 Related Work

2.1 Structured Attention and KV-Cache Reduction

One line of work improves the efficiency of long-context attention through predefined access patterns that restrict which tokens participate in attention. Longformer (Beltagy et al., 2020) combines local and global attention patterns, while StreamingLLM (Xiao et al., 2024b) maintains attention sinks and a bounded recent context for efficient streaming inference. A related line of work reduces the KV cache by retaining selected tokens for future queries. H_2O (Zhang et al., 2023) uses accumulated attention statistics, SnapKV (Li et al., 2024) selects positions based on observation-window attention, and DuoAttention (Xiao et al., 2025) distinguishes retrieval and streaming heads to reduce cache requirements. These approaches reduce attention computation or memory traffic through structured access or retention decisions.

However, the regular access patterns and low-overhead selection rules that make these approaches efficient can also constrain their ability to adapt to changing relevance across queries and contexts. HASA retains efficient structured selection through offline head-specific analysis, while dynamically organizing candidate regions and selecting relevant context at inference time.

2.2 Query-Adaptive Sparse Attention

Query-adaptive methods determine which tokens or regions to access based on the current query and context. Some approaches learn query-dependent selection explicitly. Native Sparse Attention (NSA) (Yuan et al., 2025) introduces learned compressed and selected attention branches, SeerAttention (Gao et al., 2025) trains a block-level router, and InfLLM-V2 (Zhao et al., 2026) adapts dense checkpoints to sparse retrieval through fine-tuning. SeerAttention-R (Gao et al., 2026) further extends learned routing to long-horizon reasoning. These approaches introduce additional training or learned selection components. Training-free approaches instead estimate query-dependent relevance at inference time. Quest (Tang et al., 2024) scores fixed KV pages using min–max key statistics, while InfLLM (Xiao et al., 2024a) retrieves relevant context units from external memory. MInference (Jiang et al., 2024), FlexPrefill (Lai et al., 2025), and XAttention (Xu et al., 2025) dynamically infer sparse patterns or block importance, while TokenSelect (Wu et al., 2025), SpargeAttention (Zhang et al., 2025), Tactic (Zhu et al., 2026), and Evolving Sparsity (Han et al., 2026) reduce selection cost through reuse, filtering, or adaptive sparsification. AB-Sparse (Liu et al., 2026) and DHSA (Xiong et al., 2026) further consider adaptive block granularity. Among these, UNIQUE (Deng et al., 2026) is closely related to our setting: it represents each fixed KV page using its mean key vector and a scalar dispersion term, and scores relevance using the current full-dimensional query.

However, query-dependent selection alone does not determine the appropriate resolution for each attention head. HASA further accounts for these head-specific differences by using the pretrained query–key interaction structure to adapt both the feature resolution of relevance estimation and the sequence resolution of candidate regions, achieving accurate sparse attention at low cost.

3 Methodology

We present HASA, a training-free sparse attention framework that adapts selection resolution to each head's pretrained query–key interaction structure. As illustrated in Figure 2, HASA consists of three main stages: (1) head-adaptive feature resolution, which constructs a compact response space for each attention head; (2) head-adaptive sequence resolution, which dynamically organizes candidate context into regions of varying granularity; and (3) sparse selection and efficient execution, which selects relevant regions using compact summaries and computes exact attention over their original keys and values. Together, they enable HASA to achieve accurate and efficient sparse attention.

Problem setup. Consider an attention head operating on hidden states $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_N]^\top$, where $\mathbf{h}_i \in \mathbb{R}^d$. The corresponding query and key representations are $\mathbf{q}_i = \mathbf{W}_Q \mathbf{h}_i$ and $\mathbf{k}_t = \mathbf{W}_K \mathbf{h}_t$, with head dimension d_h. Dense attention evaluates $\mathbf{q}_i^\top \mathbf{k}_t / \sqrt{d_h}$ against every preceding key, while sparse attention restricts each query to a selected subset of the candidate context.

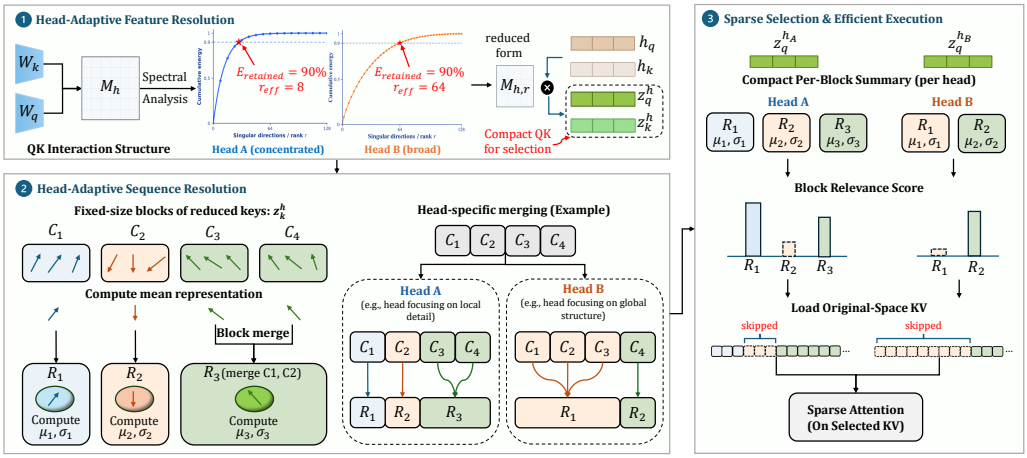


Figure 2: **Overview of HASA.** (1) Each head's pretrained query–key interaction structure determines its effective rank, yielding a compact head-specific response space. (2) In this compact response space, neighboring regions with coherent token responses are progressively merged, while heterogeneous regions remain separate. (3) Relevant regions are selected using compact region summaries, followed by exact attention over their original full-dimensional KV states.

3.1 Head-Adaptive Feature Resolution

As discussed earlier, sparse selection should account for the query–key interaction structure of each attention head without incurring the substantial overhead of estimating candidate relevance at full feature resolution. We observe that the pretrained query and key projections of each head can be combined into a single bilinear operator. For head h, the pre-softmax interaction can be written as

$$\mathbf{q}_i^\top \mathbf{k}_t = \mathbf{h}_i^\top \underbrace{(\mathbf{W}_Q^h)^\top \mathbf{W}_K^h}_{\mathbf{M}_h} \mathbf{h}_t. \tag{1}$$

The operator $\mathbf{M}_h$ depends only on pretrained model parameters and encodes the head-specific query–key interaction structure. Importantly, the interaction strength need not be distributed uniformly across feature directions: some directions contribute more strongly than others, and this distribution can vary across attention heads. This suggests that different heads may require different numbers of response directions to estimate candidate relevance. To characterize these directions, we consider the singular value decomposition $\mathbf{M}_h = \mathbf{U}_h \mathbf{\Sigma}_h \mathbf{V}_h^\top$, with singular values $\sigma_{h,1} \geq \sigma_{h,2} \geq \cdots$. The pre-softmax interaction between hidden states $\mathbf{h}_q$ and $\mathbf{h}_k$ can then be written exactly as

$$\mathbf{h}_q^\top \mathbf{M}_h \mathbf{h}_k = \left(\mathbf{\Sigma}_h^{1/2} \mathbf{U}_h^\top \mathbf{h}_q\right)^\top \left(\mathbf{\Sigma}_h^{1/2} \mathbf{V}_h^\top \mathbf{h}_k\right). \tag{2}$$

The left and right singular vectors define paired query-side and key-side response directions, while the corresponding singular values characterize their contributions to the bilinear interaction. A concentrated spectrum indicates that the head's interaction structure is dominated by relatively few directions, whereas a flatter spectrum spreads interaction strength across more directions. This motivates determining the retained feature resolution separately for each attention head.

To determine how many response directions to retain for each head, we first normalize the squared singular values as

$$p_{h,i} = \frac{\sigma_{h,i}^2}{\sum_j \sigma_{h,j}^2},$$

where $p_{h,i}$ is the fraction of spectral energy associated with direction i. We then compute the spectral entropy and its corresponding effective rank:

$$H_h = -\sum_i p_{h,i} \log p_{h,i}, \qquad r_{\mathrm{ent},h} = \lceil \exp(H_h) \rceil . \tag{3}$$

A concentrated spectrum yields a smaller effective rank, while a flatter spectrum yields a larger one. However, this entropy-based estimate alone does not guarantee that truncation preserves sufficient interaction energy. We therefore complement it with the normalized Frobenius residual of the rank-r approximation $\mathbf{M}_{h,r}$:

$$\epsilon_{F,h}(r) = \frac{\|\mathbf{M}_h - \mathbf{M}_{h,r}\|_F^2}{\|\mathbf{M}_h\|_F^2} = \frac{\sum_{i>r} \sigma_{h,i}^2}{\sum_i \sigma_{h,i}^2}. \tag{4}$$

Given an allowable residual $\epsilon_{\max}$, we define $r_{F,h} = \min\{r : \epsilon_{F,h}(r) \le \epsilon_{\max}\}$ and select the head-specific rank as

$$r_h = \max\left(r_{\mathrm{ent},h}, r_{F,h}\right).$$

This combines an estimate of the spectrum's effective dimensionality with a constraint on the energy lost through truncation, retaining the principal response directions of each head.

With r_h determined, we retain the leading singular components $\mathbf{U}_{h,r_h}$, $\mathbf{\Sigma}_{h,r_h}$, and $\mathbf{V}_{h,r_h}$ to construct paired query-side and key-side response projections:

$$\mathbf{P}_Q^h = \mathbf{\Sigma}_{h,r_h}^{1/2} \mathbf{U}_{h,r_h}^\top, \qquad \mathbf{P}_K^h = \mathbf{\Sigma}_{h,r_h}^{1/2} \mathbf{V}_{h,r_h}^\top. \tag{5}$$

For query and key hidden states, the resulting compact response representations are

$$\mathbf{z}_q^h = \mathbf{P}_Q^h \mathbf{h}_q, \qquad \mathbf{z}_k^h = \mathbf{P}_K^h \mathbf{h}_k, \tag{6}$$

which approximate the original query–key interaction through their inner product:

$$\mathbf{h}_q^\top \mathbf{M}_h \mathbf{h}_k \approx (\mathbf{z}_q^h)^\top \mathbf{z}_k^h. \tag{7}$$

These compact responses enable efficient relevance estimation in a head-adaptive feature space without operating at full feature resolution. The discarded spectrum controls the approximation error of the query–key scores, with the corresponding bound given in Appendix A. Since $\mathbf{M}_h$ depends only on pretrained model parameters, its spectral decomposition, the head-specific rank r_h, and the response projections $\mathbf{P}_Q^h$ and $\mathbf{P}_K^h$ can all be determined offline. HASA thus assigns each head a feature resolution matched to its interaction structure, rather than using the full feature dimension or a fixed rank for all heads.

3.2 Head-Adaptive Sequence Resolution

The head-specific response space constructed above also provides a basis for determining how candidate context should be organized along the sequence dimension. In this space, each token is mapped to a compact key response $\mathbf{z}_k^h$ that captures its interaction with the query-side directions retained by head h. Neighboring tokens with coherent key responses can therefore be summarized together, whereas tokens with heterogeneous responses should remain distinguishable for subsequent sparse selection. HASA uses local response coherence to construct candidate regions at varying granularities, adapting sequence resolution to each head's interaction structure.

To organize head-specific key responses into candidate regions, we first define how a group of neighboring tokens is summarized for sparse selection. To reduce candidate evaluation cost, HASA represents a candidate region C through a shared summary of its compact key responses rather than scoring each token individually. For clarity, we omit the attention-head superscript below and write $\mathbf{z}_i \in \mathbb{R}^{r_h}$ for the compact key response of token i. We define the region's mean response as

$$\boldsymbol{\mu}_C = \frac{1}{|C|} \sum_{i \in C} \mathbf{z}_i, \tag{8}$$

and its scalar dispersion as

$$\sigma_C = \sqrt{\frac{1}{|C|}\sum_{i\in C}\|\mathbf{z}_i - \boldsymbol{\mu}_C\|_2^2}. \tag{9}$$

Following the mean–dispersion scoring formulation of Deng et al. (2026), we define the relevance of region C to a compact query response $\mathbf{z}_q$ as

$$s(\mathbf{z}_q, C) = \mathbf{z}_q^\top \boldsymbol{\mu}_C + \lambda\|\mathbf{z}_q\|_2\sigma_C, \tag{10}$$

where λ controls the contribution of within-region dispersion and is fixed to 0.5 in all experiments. The mean term captures the region's average interaction with the query, while the dispersion term accounts for variation that may be suppressed by mean pooling.

After defining the region-level selection representation, we progressively merge neighboring regions whose key-response directions remain coherent. Here, comparing only region means is insufficient, as similarly oriented means may conceal heterogeneous token responses within a region. We therefore assess directional coherence across the combined region using the normalized key responses $\hat{\mathbf{z}}_i = \mathbf{z}_i/(\|\mathbf{z}_i\|_2)$. For a region C, we define

$$\mathbf{R}_C = \sum_{i\in C}\hat{\mathbf{z}}_i, \qquad \rho_C = \frac{\|\mathbf{R}_C\|_2}{|C|}, \tag{11}$$

A high ρ_C indicates concentrated response directions, whereas a low value indicates greater directional heterogeneity. This measure is related to the average pairwise agreement among token response directions and can be computed from the additive resultant $\mathbf{R}_C$; the derivation is provided in Appendix B.

We initialize the sequence with contiguous base regions of S_0 tokens ($S_0 = 8$ in our implementation) and construct larger regions through incremental merging. This bottom-up procedure allows both merge decisions and selection summaries to be updated from additive region statistics, without revisiting the constituent tokens after initialization. For each active region C, we maintain

$$n_C = |C|, \qquad \mathbf{R}_C = \sum_{i\in C}\hat{\mathbf{z}}_i, \qquad \mathbf{S}_C = \sum_{i\in C}\mathbf{z}_i, \qquad Q_C = \sum_{i\in C}\|\mathbf{z}_i\|_2^2.$$

For two adjacent regions A and B, the directional coherence of their union is

$$\rho_{A\cup B} = \frac{\|\mathbf{R}_A + \mathbf{R}_B\|_2}{n_A + n_B}. \tag{12}$$

We merge A and B when $\rho_{A\cup B} \geq \rho_{\min}$, subject to the maximum region-size constraint $n_A + n_B \leq C_{\max}$. The additive statistics also allow the selection summary of a merged region $C = A \cup B$ to be updated without revisiting its constituent tokens:

$$\boldsymbol{\mu}_C = \frac{\mathbf{S}_C}{n_C}, \qquad \sigma_C = \sqrt{\frac{Q_C}{n_C} - \|\boldsymbol{\mu}_C\|_2^2}. \tag{13}$$

Since the merge criterion and summary updates use only additive region statistics, each candidate merge requires $O(r_h)$ operations, independent of the number of tokens in the regions. This incremental procedure produces head-adaptive candidate regions at varying granularities, each equipped with a compact summary for subsequent sparse selection via equation 10.

3.3 Sparse Selection and Efficient Execution

After constructing the head-adaptive candidate regions, HASA uses their compact summaries to determine which parts of the context should be accessed for each query. For head h, the current query is projected into its head-specific response space to obtain $\mathbf{z}_q^h$. We then score each candidate region C using its maintained summary $(\boldsymbol{\mu}_C, \sigma_C)$ according to equation 10. Let I denote the number of base regions and I_h the number of candidate regions after head-adaptive merging for head h.

Table 1: Accuracy results on LongBench-Pro using Ministral-3-8B-Instruct-2512. The budget denotes the number of KV tokens available to each query during decoding; full attention is unrestricted.

Method	Budget	Overall	En	Zh	Extreme	Hard	Moderate	Easy
Full attention	–	37.70	35.88	38.96	31.13	33.72	31.60	48.86
Quest	2048	30.63	29.17	32.60	24.60	28.60	24.17	42.01
H2O	2048	33.02	31.77	34.42	27.51	31.08	26.56	43.33
InfLLM	2048	36.16	35.06	37.53	29.95	32.90	29.84	**47.98**
UNIQUE	2048	37.11	35.57	38.15	31.28	33.18	**31.03**	47.66
HASA (Ours)	2048	**37.62**	**36.90**	**38.28**	**32.80**	**34.10**	31.01	47.25
Quest	1024	27.18	25.83	28.76	21.53	22.69	21.62	38.71
H2O	1024	31.61	30.59	33.16	26.93	29.15	25.43	41.87
InfLLM	1024	35.42	34.77	36.68	29.21	32.58	28.60	48.12
UNIQUE	1024	36.72	35.77	37.98	31.02	32.78	30.75	48.34
HASA (Ours)	1024	**37.53**	**36.66**	**38.74**	**32.17**	**33.83**	**32.27**	**48.63**
Quest	512	21.72	21.09	21.84	18.87	16.18	17.61	29.31
H2O	512	29.04	27.96	30.44	23.61	28.15	22.20	39.13
InfLLM	512	34.99	33.82	36.21	28.96	32.38	27.31	46.74
UNIQUE	512	36.58	**35.22**	38.23	30.86	33.75	30.37	47.67
HASA (Ours)	512	**37.43**	35.02	**40.13**	**30.91**	**34.27**	**32.94**	**48.46**

Scoring these regions requires $O(I_h r_h)$ operations per query, compared with $O(Id)$ for scoring the base regions at full feature resolution. Since merging does not increase the number of regions and $r_h \leq d$, head-adaptive sequence and feature resolutions jointly reduce the region-scoring cost. This produces a query-dependent ranking of candidate regions without computing full-dimensional attention scores for their constituent tokens.

HASA selects the top-K regions according to these scores and additionally includes the local context. Let $\mathcal{C}_q^h$ denote the selected candidate regions and $\mathcal{L}_q$ the local token positions. Together, they define the sparse attention support:

$$\mathcal{I}_q^h = \left(\cup_{C \in \mathcal{C}_q^h} C\right) \cup \mathcal{L}_q. \tag{14}$$

Once $\mathcal{I}_q^h$ is determined, HASA retrieves the original full-dimensional keys and values at the selected positions and computes exact attention over this sparse region. By scoring compact region summaries rather than individual full-dimensional keys, HASA keeps the selection lightweight. By accessing only the selected KV, it also reduces memory traffic and attention computation.

Overall, HASA combines head-adaptive feature and sequence resolutions to identify relevant context at an appropriate granularity, while using the original query, keys, and values for attention over the selected tokens. This design preserves attention accuracy while reducing inference latency.

4 Experiments

4.1 Experimental Setup

We evaluate HASA on **LongBench-Pro** (Chen et al., 2026) and **RULER-32K** (Hsieh et al., 2024), covering realistic multilingual long-context tasks and controlled synthetic tasks, respectively. Experiments are conducted on **Ministral-3-8B-Instruct-2512** and **Qwen3-8B** to assess generalization across model families. We compare HASA with four training-free sparse-attention baselines: **H_2O** (Zhang et al., 2023), **Quest** (Tang et al., 2024), **InfLLM** (Xiao et al., 2024a), and **UNIQUE** (Deng et al., 2026), together with **dense attention** as the full-context reference. Sparse methods are evaluated under matched per-query KV budgets across multiple sparsity levels. We report overall, language-specific, and difficulty-level scores on LongBench-Pro and the average score on RULER-32K. Benchmark details, baseline configurations, and evaluation protocols are in Appendix C.1.

4.2 Results and Analysis

Accuracy. Tables 1 and 2 compare HASA with the baselines under matched KV budgets on Ministral-3-8B-Instruct-2512. Across both benchmarks, HASA achieves the highest overall scores among the evaluated sparse methods, with scores close to those of dense attention.

Table 2: Accuracy results on RULER-32K using Ministral-3-8B-Instruct-2512. The budget denotes the number of KV tokens available to each query during decoding; full attention is unrestricted.

Method	Budget	SG1	SG2	SG3	MK1	MK2	MK3	MV	MQ	VT	CWE	FWE	QA1	QA2	Avg.
Full attention	–	100.00	99.60	99.60	99.40	100.00	99.80	98.55	99.05	99.96	88.46	93.27	75.80	57.40	93.15
Quest	2048	**100.00**	98.80	97.00	99.20	99.80	97.60	99.45	99.30	99.88	77.80	90.67	75.00	57.20	91.67
H2O	2048	**100.00**	99.80	31.00	99.60	97.80	1.80	92.65	65.20	98.40	42.96	90.80	74.40	56.80	73.17
InfLLM	2048	**100.00**	94.00	85.20	95.20	99.80	96.40	93.00	88.45	94.08	22.10	83.80	74.60	57.20	83.37
UNIQUE	2048	**100.00**	99.80	99.20	99.60	**100.00**	99.80	**99.00**	**99.15**	99.96	82.28	**91.07**	75.60	**57.40**	92.53
HASA (Ours)	2048	**100.00**	**100.00**	**100.00**	**100.00**	99.00	**99.80**	98.51	**99.15**	**100.00**	**83.86**	90.43	**76.53**	**57.40**	**92.67**
Quest	1024	**100.00**	98.80	97.00	99.16	99.76	86.45	**99.44**	**99.29**	99.88	71.61	90.53	74.86	56.72	90.27
H2O	1024	99.90	98.85	17.94	98.72	90.67	0.92	67.59	41.44	82.32	32.39	79.12	74.40	56.31	64.66
InfLLM	1024	96.52	89.68	80.78	92.23	99.16	69.57	89.52	84.02	90.67	16.35	79.01	74.41	56.74	78.36
UNIQUE	1024	**100.00**	99.80	99.20	99.60	**100.00**	**99.80**	99.00	99.15	99.96	75.38	**90.96**	75.16	**57.56**	91.97
HASA (Ours)	1024	**100.00**	**100.00**	**100.00**	**100.00**	99.01	99.79	98.51	99.15	**100.00**	**78.39**	90.27	**76.09**	**57.56**	**92.21**
Quest	512	99.40	94.60	91.00	97.20	95.80	53.00	96.55	96.65	99.00	53.24	87.60	74.00	55.80	84.14
H2O	512	99.80	98.00	2.60	97.60	84.60	0.00	44.45	14.65	66.88	20.94	68.33	74.40	55.80	56.00
InfLLM	512	92.60	86.00	76.00	88.60	98.40	43.80	85.60	79.40	87.52	10.12	74.40	74.20	56.20	73.30
UNIQUE	512	**100.00**	99.80	99.20	**99.80**	**100.00**	**99.60**	99.55	**99.30**	99.96	62.90	**88.67**	73.60	**57.80**	90.78
HASA (Ours)	512	**100.00**	**100.00**	**100.00**	**99.80**	**100.00**	99.20	**100.00**	**100.00**	**100.00**	**66.37**	87.73	**74.54**	57.40	**91.16**

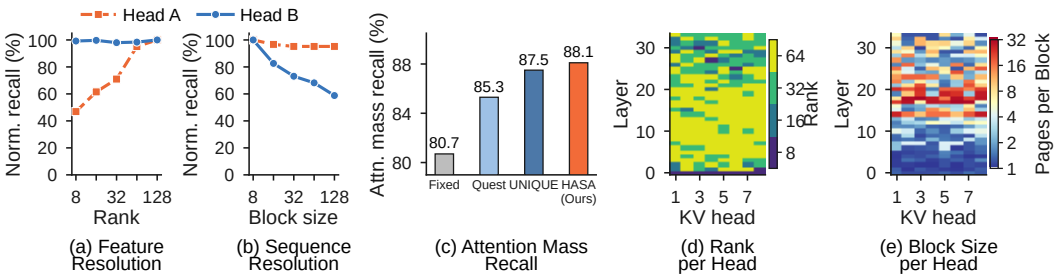
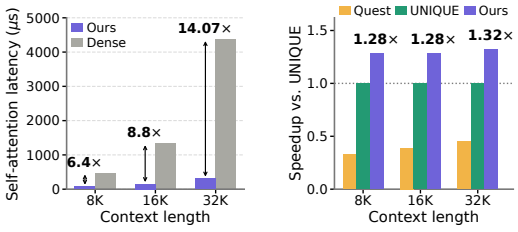


Figure 3: **Head-specific resolution and attention recall.** (a–b) High recall is achievable by adjusting feature rank and block size, with different heads requiring different resolutions. (c) Adaptive resolution improves normalized attention recall over fixed configurations across RULER-32K tasks. (d–e) HASA selects varying feature ranks and sequence granularities across layers and KV heads, confirming head-specific adaptation.

On LongBench-Pro, HASA achieves **37.62**, **37.53**, and **37.43** at KV budgets of 2048, 1024, and 512, respectively, compared with 37.11, 36.72, and 36.58 for UNIQUE. The advantage becomes more pronounced as the budget decreases: at 512 tokens, HASA remains within only 0.27 points of dense attention (37.70). The gains also extend across language and difficulty breakdowns. At a 1024-token budget, HASA scores **36.66** on English and **38.74** on Chinese, with strong performance on the Extreme, Hard, and Moderate subsets.



(a) Comparison with dense attention.

(b) Comparison with sparse baselines.

Figure 4: Self-attention efficiency at increasing context lengths. **(a)** Our method substantially reduces latency compared with dense attention. **(b)** Our method achieves higher self-attention speedup than Quest and UNIQUE.

On RULER-32K, HASA obtains average scores of **92.67**, **92.21**, and **91.16** at the same budgets, compared with 92.53, 91.97, and 90.78 for UNIQUE. HASA also maintains strong performance on retrieval and multi-key tasks, including challenging settings such as CWE and QA1. Reducing the budget from 2048 to 512 lowers HASA's average score by only 1.51 points. Together, these results suggest that head-adaptive selection helps retain relevant context when the available KV budget becomes increasingly restrictive. Similar trends are observed on **Qwen3-8B** (Appendix C.2).

Adaptive Resolution. Figure 3 examines the effectiveness of head-adaptive feature and sequence resolutions. Figures 3(a–b) reveal distinct resolution requirements across heads: Head B retains **99.2%** recall at rank 8, while Head A requires rank 64 to reach **95.1%**; conversely, at block size

128, Head A maintains **95.2%** recall whereas Head B drops to **59%**. These contrasting requirements motivate joint adaptation across both dimensions. Figure 3(c) evaluates the resulting selection fidelity using attention-mass recall under matched KV budgets, where the full dense-attention mass is 100%. For the fixed-resolution reference, we disable both forms of head-wise adaptation: all heads use a shared feature rank equal to the average of HASA's head-specific ranks and fixed 8-token pages without adaptive merging. HASA achieves **88.1%** attention-mass recall, compared with **80.7%** for this fixed configuration, an improvement of **7.4 percentage points. It also exceeds Quest (85.3%) and UNIQUE (87.5%), indicating that adapting selection resolution better preserves the context emphasized by dense attention.** Finally, Figures 3(d–e) visualize the varying feature and sequence resolutions across layers and KV heads, confirming HASA's adaptation to the underlying head interaction structure.

Efficiency. Figure 4 reports self-attention latency across increasing context lengths. HASA achieves speedups of **6.4**×, **8.8**×, and **14.07**× over dense attention at 8K, 16K, and 32K, respectively, demonstrating increasing benefits from sparse execution at longer contexts. HASA also improves over the training-free sparse-attention baselines: compared with UNIQUE, it achieves **1.28**×, **1.28**×, and **1.32**× speedups at the same context lengths. This additional gain over a sparse baseline is consistent with HASA's use of compact head-specific responses and region summaries to reduce query-dependent selection cost, with the advantage becoming more pronounced for longer context.

Ablation Study. Table 3 isolates the contributions of adaptive feature and sequence resolution at a 1024-token KV budget on RULER-32K. The fully fixed configuration follows the fixed-resolution reference in Figure 3(c), using a shared feature rank and fixed 8-token pages without adaptive merging. Starting from the fully fixed configuration, adaptive feature resolution improves accuracy from 91.18% to 91.42% and normalized attention recall from 91.60% to 93.40%, while adaptive sequence resolution achieves 91.50% accuracy and 94.00% recall. Combining both yields the highest accuracy and normalized recall, reaching 92.21% and 100%, respectively, corresponding to gains of 1.03 and 8.40 percentage points over the fully fixed configuration. These results show that the two adaptations are complementary: each individually improves preservation of dense-attention relevance, while their combination further improves both selection fidelity and downstream accuracy.

Table 3: Ablation of adaptive feature and sequence resolution at 1024 KV budget, RULER-32K.

Feature	Sequence	Acc. (%)	Norm. Recall (%)
Fixed	Fixed	91.18	91.60
Adaptive	Fixed	91.42	93.40
Fixed	Adaptive	91.50	94.00
Adaptive	**Adaptive**	**92.21**	**100.00**

5 Conclusion

We presented **HASA**, a training-free sparse attention framework that adapts feature and sequence resolutions to the pretrained query–key interaction structure of each attention head. HASA determines head-specific feature ranks through spectral analysis, retaining the dominant interaction directions. It further constructs candidate regions according to the coherence of compact key responses, adapting sequence granularity to each head's interaction structure. HASA then performs exact attention over the selected full-dimensional keys and values. Across benchmarks, KV budgets, and model backbones, HASA improves long-context accuracy over training-free sparse-attention baselines while remaining close to dense attention. Quantitatively, HASA captures **88.1%** of dense-attention mass, improves accuracy over UNIQUE by up to **3.47 points** on RULER and **2.57 points** on LongBench-Pro, and accelerates self-attention by up to **14.07**× over dense attention and **1.32**× over UNIQUE.

Limitations. HASA characterizes each attention head using the pretrained bilinear query–key operator $\mathbf{M}_h = (\mathbf{W}_Q^h)^\top \mathbf{W}_K^h$, enabling its response structure to be determined offline. However, this input-independent characterization does not capture the full runtime sensitivity of attention, including the effects of softmax normalization, value representations, and downstream computation. Input-dependent information, such as local Jacobians, could provide a more faithful characterization but would incur additional inference overhead. Future work could explore lightweight input-dependent corrections that improve selection fidelity while preserving HASA's efficiency.

AI USAGE STATEMENT

In this work, Generative AI was used to improve the grammar, clarity, and readability of the author-written text, and LaTeX formatting issues. Coding assistants were also used for implementation tasks, such as generating plots, improving code organization and structure, and assisting with the routine implementation. Generative AI was not used to develop research ideas, formulate hypotheses, design experiments, analyze experimental results, or draw scientific conclusions. All AI-assisted text and code were carefully reviewed by the authors, and generated code was verified and tested for correctness before use. The authors take full responsibility for the final content of this work, including all text, claims, experimental results, code, and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work uses publicly available models and benchmarks and involves no human subjects, personally identifiable information, or new data collection.

REFERENCES

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.

Ziyang Chen, Xing Wu, Junlong Jia, Chaochen Gao, Qi Fu, Debing Zhang, and Songlin Hu. Longbench pro: A more realistic and comprehensive bilingual long-context evaluation benchmark, 2026. URL https://arxiv.org/abs/2601.02872.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv preprint arXiv:2507.06261*, 2025. URL https://arxiv.org/abs/2507.06261.

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. Adaptively sparse transformers. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2174–2184, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1223. URL https://aclanthology.org/D19-1223/.

Tri Dao. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/98ed250b203d1ac6b24bbcf263e3d4a7-Abstract-Conference.html.

Keqi Deng, Shaoshi Ling, Ruchao Fan, and Jinyu Li. Unique: Universal top-k sparse attention for training-free inference and sparsity-aware training, 2026. URL https://arxiv.org/abs/2605.27740.

Yizhao Gao, Zhichen Zeng, Dayou Du, Shijie Cao, Peiyuan Zhou, Jiaxing Qi, Junjie Lai, Hayden So, Ting Cao, Fan Yang, and Mao Yang. SeerAttention: Self-distilled Attention Gating for Efficient Long-context Prefilling. In *Advances in Neural Information Processing Systems*, volume 38, pp. 62230–62253, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/50e9dbc4ab68d94f15261ddc26c8ca2b-Abstract-Conference.html.

Yizhao Gao, Shuming Guo, Shijie Cao, Yuqing Xia, Yu Cheng, Lei Wang, Lingxiao Ma, Yutao Sun, Tianzhu Ye, Li Dong, Hayden So, Yu Hua, Ting Cao, Fan Yang, and Mao Yang. Sparse attention adaptation for long reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 111697–111714, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/b56d827a2b8433517e722e0272c7f464-Paper-Conference.pdf.

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

Ruizi Han, Miao Zhang, Ziyue Qiao, and Liqiang Nie. Evolving Sparsity: Leveraging Token Importance Dynamics for Efficient LLM Decoding with Sparse Attention. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11554–11566. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.530. URL https://aclanthology.org/2026.acl-long.530/.

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?, 2024. URL https://arxiv.org/abs/2404.06654.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. MInference 1.0: Accelerating Pre-filling for Long-Context LLMs via Dynamic Sparse Attention. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5dfbe6f5671e82c76841ba687a8a9ecb-Abstract-Conference.html.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626. Association for Computing Machinery, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. FlexPrefill: A Context-Aware Sparse Attention Mechanism for Efficient Long-Sequence Inference. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/03645743ea35690f30d795d6bac149a5-Abstract-Conference.html.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM Knows What You are Looking for Before Generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 22947–22970, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/28ab418242603e0f7323e54185d19bde-Abstract-Conference.html.

Di Liu, Ruitian Wang, Chen Chen, Mingliang Gong, Yongjie Yuan, Han Zhao, Yu Feng, Quan Chen, and Minyi Guo. AB-Sparse: Sparse Attention with Adaptive Block Size for Accurate and Efficient Long-Context Inference. *arXiv preprint arXiv:2605.12110*, 2026. URL https://arxiv.org/abs/2605.12110.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan

Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently Scaling Transformer Inference. In *Proceedings of Machine Learning and Systems*, volume 5, 2023. URL https://proceedings.mlsys.org/paper_files/paper/2023/hash/c4be71ab8d24cdfb45e3d06dbfca2780-Abstract-mlsys2023.html.

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

Wei Wu, Zhuoshi Pan, Kun Fu, Chao Wang, Liyi Chen, Yunchu Bai, Tianfu Wang, Zheng Wang, and Hui Xiong. TokenSelect: Efficient Long-Context Inference and Length Extrapolation for LLMs via Dynamic Token-Level KV Cache Selection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21264–21281. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1079. URL https://aclanthology.org/2025.emnlp-main.1079/.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. InfLLM: Training-Free Long-Context Extrapolation for LLMs with an Efficient Context Memory. In *Advances in Neural Information Processing Systems*, volume 37, 2024a. doi: 10.52202/079017-3801. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient Streaming Language Models with Attention Sinks. In *International Conference on Learning Representations*, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5e5fd18f863cbe6d8ae392a93fd271c9-Abstract-Conference.html.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. DuoAttention: Efficient Long-Context LLM Inference with Retrieval and Streaming Heads. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5c1ddd2e59df46fd2aa85c833b1b36ed-Abstract-Conference.html.

Siheng Xiong, Joe Zou, Faramarz Fekri, and Yae Jee Cho. Long-context modeling with dynamic hierarchical sparse attention for memory-constrained llm inference. In *Forty-third International Conference on Machine Learning*, 2026.

Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. XAttention: Block Sparse Attention with Antidiagonal Scoring. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 69819–69831. PMLR, 2025. URL https://proceedings.mlr.press/v267/xu25ag.html.

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. URL https://aclanthology.org/2025.acl-long.1126/.

Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. SpargeAttention: Accurate and Training-free Sparse Attention Accelerating Any Model Inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhang25ch.html.

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, pp. 34661–34710, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/6ceefa7b15572587b78ecfcebb2827f8-Abstract-Conference.html.

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. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/716e60181e1782f5bd0a629033f85310-Abstract-Conference.html.

Kan Zhu, Tian Tang, Qinyu Xu, Zhan Jin, Yile Gu, Zhichen Zeng, Rohan Kadekodi, Liangyu Zhao, Ang Li, Arvind Krishnamurthy, and Baris Kasikci. Tactic: Adaptive Sparse Attention with Clustering and Distribution Fitting for Long-Context LLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/33f94d79acf71051d6a27f4d8889e20e-Abstract-Conference.html.

A THEORETICAL GROUNDING OF SPECTRAL ROUTING

Approximation of region scores. HASA ranks candidate regions using the mean–dispersion score in equation 10, rather than scoring individual keys. We therefore examine how spectral truncation affects this region-level score.

For a fixed region C, let $S_C^{(p)}$ and $S_C^{(r)}$ denote the scores computed from the full and rank-r spectral response representations, respectively, using the same mean–dispersion formulation. Here $p = \mathrm{rank}(M_h)$, so the full response retains all nonzero singular directions. Define the mean and dispersion of the original key hidden states within C as

$$\bar{x}_C = \frac{1}{|C|}\sum_{i\in C} x_{k,i}, \qquad v_C = \left(\frac{1}{|C|}\sum_{i\in C}\|x_{k,i} - \bar{x}_C\|_2^2\right)^{1/2}.$$

Lemma 1 (Region-score approximation)**.** *Let $\gamma = \sigma_{h,r+1}$ and $L = \sigma_{h,1}$. For any fixed query and region C, the mean–dispersion scores satisfy*

$$\left|S_C^{(p)} - S_C^{(r)}\right| \le \gamma\|x_q\|_2\|\bar{x}_C\|_2 + \lambda\sqrt{2L\gamma}\,\|x_q\|_2 v_C.$$

Proof. Partition each full spectral response into its retained and discarded components. Because these components occupy orthogonal coordinates, the difference between the full and truncated mean terms is

$$\begin{aligned}\left|(\mathbf{z}_q^{(p)})^\top \boldsymbol{\mu}_C^{(p)} - (\mathbf{z}_q^{(r)})^\top \boldsymbol{\mu}_C^{(r)}\right| &= \left|x_q^\top (M_h - M_{h,r})\bar{x}_C\right| \\ &\le \gamma\|x_q\|_2\|\bar{x}_C\|_2.\end{aligned}$$

It remains to bound the dispersion contribution. Let $a_p = \|\mathbf{z}_q^{(p)}\|_2$ and $a_r = \|\mathbf{z}_q^{(r)}\|_2$, and let $\sigma_C^{(p)}$ and $\sigma_C^{(r)}$ denote the corresponding response dispersions. Orthogonality of the retained and discarded components gives

$$a_p^2 - a_r^2 \le \gamma\|x_q\|_2^2, \qquad (\sigma_C^{(p)})^2 - (\sigma_C^{(r)})^2 \le \gamma v_C^2.$$

Moreover,

$$a_r^2 \le L\|x_q\|_2^2, \qquad (\sigma_C^{(p)})^2 \le L v_C^2.$$

Since $a_p \ge a_r$ and $\sigma_C^{(p)} \ge \sigma_C^{(r)}$,

$$\begin{aligned}&\left|a_p\sigma_C^{(p)} - a_r\sigma_C^{(r)}\right|^2 \\ &\quad\le a_p^2(\sigma_C^{(p)})^2 - a_r^2(\sigma_C^{(r)})^2 \\ &\quad= (a_p^2 - a_r^2)(\sigma_C^{(p)})^2 + a_r^2\left((\sigma_C^{(p)})^2 - (\sigma_C^{(r)})^2\right) \\ &\quad\le 2L\gamma\|x_q\|_2^2 v_C^2.\end{aligned}$$

Multiplying by λ and combining the bounds for the mean and dispersion terms proves the result. □

Preservation of block selection. A bounded region-score error does not guarantee identical ranking when candidate regions have nearly identical scores. However, a sufficient margin at the selection boundary preserves the selected blocks.

Corollary 1 (Top-K block selection preservation)**.** *Consider a fixed query and a fixed collection of candidate regions $\mathcal{C} = \{C_1, \ldots, C_n\}$. Let $S_j^{(p)}$ and $S_j^{(r)}$ be their full-rank and rank-r mean–dispersion scores, respectively, and let*

$$\epsilon = \max_{1\le j\le n}\left[\gamma\|x_q\|_2\|\bar{x}_{C_j}\|_2 + \lambda\sqrt{2L\gamma}\,\|x_q\|_2 v_{C_j}\right].$$

If the K-th and $(K+1)$-th largest full-rank region scores satisfy

$$S_{(K)}^{(p)} - S_{(K+1)}^{(p)} > 2\epsilon,$$

then the top-K region sets under the full-rank and rank-r scores are identical.

Proof. Lemma 1 gives $|S_j^{(p)} - S_j^{(r)}| \le \epsilon$ for every candidate region. For any region C_i in the full-rank top-K set and any region C_j outside it,

$$S_i^{(r)} \ge S_{(K)}^{(p)} - \epsilon, \qquad S_j^{(r)} \le S_{(K+1)}^{(p)} + \epsilon.$$

The assumed margin therefore implies $S_i^{(r)} > S_j^{(r)}$ for every such pair. Hence no region outside the full-rank top-K set can displace a selected region. □

Together, these results clarify the theoretical role of spectral truncation in HASA. The discarded singular spectrum bounds both the underlying query–key interaction error and the approximation error of the mean–dispersion region scores used for routing. Under a sufficient selection margin, this bound further guarantees preservation of top-K selection over a fixed collection of candidate regions. The analysis isolates the effect of feature truncation and does not guarantee identical region partitions when feature rank changes. After selection, HASA computes exact attention over the selected original full-dimensional KV states.

B Analysis of the Directional Purity Criterion

This section provides additional analysis of the directional-purity criterion introduced in Section 3.2. In particular, we establish its relationship to pairwise directional agreement and contrast it with directly comparing neighboring block representatives.

B.1 Relation to Pairwise Directional Agreement

Recall that for a block C with $n_C = |C|$ normalized nonzero token responses $\hat{\mathbf{z}}_i$, directional purity is defined as

$$\rho_C = \frac{1}{n_C} \left\| \sum_{i \in C} \hat{\mathbf{z}}_i \right\|_2 .$$

Squaring both sides gives

$$\rho_C^2 = \frac{1}{n_C^2} \left\| \sum_{i \in C} \hat{\mathbf{z}}_i \right\|_2^2 \tag{15}$$

$$= \frac{1}{n_C^2} \left(\sum_{i \in C} \|\hat{\mathbf{z}}_i\|_2^2 + 2 \sum_{i<j} \hat{\mathbf{z}}_i^\top \hat{\mathbf{z}}_j \right). \tag{16}$$

Since $\|\hat{\mathbf{z}}_i\|_2 = 1$, this reduces to

$$\rho_C^2 = \frac{1}{n_C^2} \left(n_C + 2 \sum_{i<j} \hat{\mathbf{z}}_i^\top \hat{\mathbf{z}}_j \right). \tag{17}$$

Because the representations are normalized, $\hat{\mathbf{z}}_i^\top \hat{\mathbf{z}}_j$ is their cosine similarity. For $n_C \ge 2$, let the average pairwise cosine agreement within C be

$$\bar{c}_C = \frac{2}{n_C(n_C-1)} \sum_{i<j} \hat{\mathbf{z}}_i^\top \hat{\mathbf{z}}_j .$$

Substituting into Eq. equation 17 yields

$$\rho_C^2 = \frac{1}{n_C} + \frac{n_C - 1}{n_C} \bar{c}_C, \tag{18}$$

and equivalently,

$$\bar{c}_C = \frac{n_C \rho_C^2 - 1}{n_C - 1}. \tag{19}$$

Thus, ρ_C is directly related to the average pairwise directional agreement of all token representations within the block. Importantly, it can be computed from a single resultant vector, avoiding explicit evaluation of the $\mathcal{O}(n_C^2)$ token pairs.

B.2 Why Block-Mean Cosine Is Insufficient

An alternative merge criterion is to compare neighboring block representatives using $\cos(\boldsymbol{\mu}_A, \boldsymbol{\mu}_B)$. This quantity, however, depends only on the two means and contains no information about the distribution of directions within either block.

To see this, consider two blocks whose token representations are internally dispersed but whose mean directions are similar. Their block means may satisfy $\cos(\boldsymbol{\mu}_A, \boldsymbol{\mu}_B) \approx 1$, even though the tokens within $A \cup B$ exhibit weak directional agreement. A cosine-based merge would therefore treat the blocks as compatible solely because their first-order summaries align.

Directional purity instead evaluates

$$\rho_{A\cup B} = \frac{\|\mathbf{R}_A + \mathbf{R}_B\|_2}{n_A + n_B},$$

which depends on the accumulated directions of all tokens in the proposed union. As shown by equation 19, a high $\rho_{A\cup B}$ implies high aggregate pairwise agreement within the merged region, rather than merely alignment between two block means.

B.3 Exact Update of Pooled Statistics

The additive statistics maintained during merging also allow the pooled quantities used for block scoring to be recovered without revisiting individual tokens. For a block C, let

$$\mathbf{S}_C = \sum_{i\in C} \mathbf{z}_i, \qquad Q_C = \sum_{i\in C} \|\mathbf{z}_i\|_2^2.$$

The mean follows immediately as

$$\boldsymbol{\mu}_C = \frac{\mathbf{S}_C}{n_C}.$$

Using the identity

$$\frac{1}{n_C}\sum_{i\in C} \|\mathbf{z}_i - \boldsymbol{\mu}_C\|_2^2 = \frac{Q_C}{n_C} - \|\boldsymbol{\mu}_C\|_2^2,$$

the scalar dispersion used by the block-scoring function is

$$\sigma_C = \sqrt{\frac{Q_C}{n_C} - \|\boldsymbol{\mu}_C\|_2^2}.$$

Since $\mathbf{S}_C$, Q_C, and $\mathbf{R}_C$ are additive under block merging, both the purity criterion and pooled scoring statistics can be updated using only block-level quantities.

C Experimental Setup and Additional Analysis

C.1 Evaluation Setup

Benchmarks. We evaluate HASA on two complementary long-context benchmarks: **LongBench-Pro** (Chen et al., 2026) and **RULER-32K** (Chen et al., 2026). LongBench-Pro contains 1,500 English and Chinese examples spanning 11 primary and 25 secondary tasks, with input lengths extending to 256K tokens. It evaluates long-context understanding across different languages and difficulty levels. RULER-32K provides a controlled evaluation at a 32K context length, covering retrieval, multi-hop tracing, aggregation, and question-answering tasks. Together, the two benchmarks test both realistic long-context reasoning and controlled retrieval over extended contexts.

Baselines. We compare HASA against four representative training-free sparse-attention methods. $\mathbf{H_2O}$ retains tokens according to accumulated historical attention importance; **Quest** performs query-aware retrieval over fixed KV pages using per-page min/max key statistics; **InfLLM** retrieves relevant context units from a long-term memory representation; and **UNIQUE**, the closest baseline to our setting, represents each fixed KV page using its mean key vector and scalar dispersion for query-dependent selection. We also report **dense attention** as the full-context reference. All sparse methods are evaluated without additional model training or fine-tuning.

Table 4: Accuracy results on LongBench-Pro using Qwen3-8B. The KV budget denotes the number of KV tokens available to each query during decoding; full attention is unrestricted.

Method	Budget	Overall	En	Zh	Extreme	Hard	Moderate	Easy
Full attention	–	37.13	35.92	38.34	27.49	35.21	31.44	46.86
Quest	2048	35.02	33.49	36.55	25.62	33.29	29.33	44.51
UNIQUE	2048	36.12	35.03	37.22	27.15	33.49	29.85	45.99
HASA (Ours)	2048	**36.64**	**35.77**	**37.50**	**27.62**	**34.32**	**30.06**	**46.53**
Quest	1024	34.31	33.14	35.47	24.71	**33.24**	27.94	43.92
UNIQUE	1024	35.16	**34.17**	36.15	**26.62**	32.46	28.41	44.99
HASA (Ours)	1024	**35.63**	33.85	**37.42**	26.27	33.02	**29.73**	**45.59**
Quest	512	32.87	31.36	34.37	23.69	28.77	25.24	44.13
UNIQUE	512	34.49	**33.41**	35.56	24.88	**33.37**	27.74	**44.31**
HASA (Ours)	512	**34.84**	32.41	**37.27**	**26.60**	32.15	**28.74**	44.17

Table 5: Accuracy results on RULER-32K using Qwen3-8B. The KV budget denotes the number of KV tokens available to each query during decoding; full attention is unrestricted.

Method	Budget	SG1	SG2	SG3	MK1	MK2	MK3	MV	MQ	VT	CWE	FWE	QA1	QA2	Avg.
Dense	–	100.00	100.00	100.00	100.00	97.00	95.00	99.75	99.75	99.80	84.10	94.67	75.00	50.00	91.93
Quest	2048	**100.00**	**100.00**	**100.00**	99.00	**98.00**	78.00	**100.00**	99.00	98.20	58.60	90.67	**75.00**	55.00	88.57
UNIQUE	2048	**100.00**	**100.00**	**100.00**	99.00	**98.00**	79.00	**100.00**	99.25	98.20	59.50	90.67	**75.00**	**58.00**	88.97
HASA (Ours)	2048	**100.00**	**100.00**	**100.00**	**100.00**	97.00	**84.00**	99.75	**99.75**	**99.80**	**71.40**	**93.67**	74.00	56.00	**90.41**
Quest	1024	**100.00**	**100.00**	**100.00**	**100.00**	96.00	44.00	99.75	99.25	**100.00**	**61.00**	**93.67**	71.00	53.00	85.97
UNIQUE	1024	**100.00**	**100.00**	**100.00**	99.00	**97.00**	62.00	98.75	**99.75**	96.80	48.10	88.67	72.00	**56.00**	86.01
HASA (Ours)	1024	**100.00**	**100.00**	**100.00**	97.00	88.00	**82.00**	**100.00**	96.75	96.80	56.10	88.67	**73.00**	50.00	**86.79**
Quest	512	**100.00**	**100.00**	**100.00**	**100.00**	90.00	7.00	96.50	97.75	**99.60**	**44.90**	**95.33**	70.00	49.00	80.78
UNIQUE	512	**100.00**	**100.00**	98.00	99.00	**97.00**	41.00	97.75	**99.00**	95.60	35.30	87.67	73.00	**55.00**	82.95
HASA (Ours)	512	**100.00**	**100.00**	99.00	96.00	81.00	**66.00**	**98.75**	**99.00**	95.20	43.20	81.33	**74.00**	52.00	**83.50**

Evaluation Protocol. Experiments use Ministral-3-8B-Instruct-2512 for the main evaluation, with Qwen3-8B used as an additional backbone in Appendix C.2. This allows us to assess whether the proposed selection mechanism generalizes across different model architectures. We report the individual task scores and their average. Sparse methods are compared under matched per-query KV budgets of **512**, **1024**, and **2048** tokens during decoding, while dense attention accesses the full available KV context. We follow the corresponding benchmark evaluation protocols and use the same model, input, decoding, and KV-budget settings across sparse methods to ensure a matched comparison. Unless otherwise specified, HASA uses a base region size of $S_0 = 8$, $\lambda = 0.5$, $\epsilon_{\max} = 0.10$, $\rho_{\min} = 0.70$, and $C_{\max} = 512$.

C.2 Additional Backbone Evaluation

To further evaluate the effectiveness of HASA across model backbones, we conduct additional experiments on Qwen3-8B using the same evaluation protocol and matched KV budgets. As shown in Tables 4 and 5, HASA achieves the highest overall scores among the evaluated sparse methods at every KV budget on both LongBench-Pro and RULER-32K.

On LongBench-Pro, HASA achieves overall scores of **36.64**, **35.63**, and **34.84** at KV budgets of 2048, 1024, and 512, respectively, compared with 36.12, 35.16, and 34.49 for UNIQUE. The gains therefore persist as the available KV budget is reduced, with improvements of 0.52, 0.47, and 0.35 points over the strongest sparse baseline. The language- and difficulty-level breakdowns show that the advantage is not restricted to a single subset: HASA improves substantially on Chinese at the lower budgets and remains competitive across the remaining categories. A similar pattern is observed on RULER-32K. HASA obtains average scores of **90.41**, **86.79**, and **83.50** at budgets of 2048, 1024, and 512, respectively, outperforming UNIQUE by 1.44, 0.78, and 0.55 points. The improvement is particularly visible on several difficult multi-key and retrieval-sensitive tasks, while performance on already saturated retrieval tasks remains comparable across methods. Importantly, HASA continues to retain the highest overall score as the KV budget becomes more restrictive.

Table 6: Hyperparameter sensitivity on RULER-32K at a 1024-token KV budget. Each hyperparameter is varied independently while the remaining parameters are fixed to their best-performing values: $\epsilon_{\max} = 0.10$, $\rho_{\min} = 0.70$, and $C_{\max} = 512$.

Hyperparameter	Value	Accuracy (%)
$\epsilon_{\max}$	0.05	92.21
	0.10	**92.21**
	0.15	91.98
	0.20	91.04
$\rho_{\min}$	0.50	90.82
	0.60	91.73
	0.70	**92.21**
	0.80	92.16
$C_{\max}$	128	92.16
	256	92.17
	512	**92.21**
	1024	91.42

The consistent improvements across model backbones, benchmarks, and KV budgets demonstrate the broad effectiveness and generalizability of HASA. By deriving feature ranks and sequence granularity from each head's pretrained query–key interaction structure, HASA adapts its sparse selection to different models rather than relying on a fixed resolution configuration.

C.3 Hyperparameter Analysis

Table 6 examines the sensitivity of HASA to the three hyperparameters controlling feature and sequence resolution. We use RULER-32K and the intermediate 1024-token KV budget as a representative setting, varying each hyperparameter independently while holding the others fixed. Overall, performance remains stable over a reasonably broad range around the default configuration.

For the spectral residual threshold $\epsilon_{\max}$, HASA achieves the same reported accuracy (92.21%) at 0.05 and 0.10. We adopt 0.10 as the default because its more permissive residual constraint can reduce the retained feature rank and selection overhead without sacrificing accuracy in this setting. Increasing the threshold to 0.15 causes only a slight decline to 91.98%, whereas a larger threshold of 0.20 reduces accuracy to 91.04%. This indicates that excessively aggressive spectral truncation can discard interaction directions that remain useful for sparse selection.

The sequence-resolution parameters show a similar trade-off. Lowering the purity threshold $\rho_{\min}$ permits more aggressive aggregation of neighboring regions and reduces accuracy from 92.21% at $\rho_{\min} = 0.70$ to 90.82% at 0.50. Increasing the threshold to 0.80 produces nearly identical performance (92.16%), suggesting that $\rho_{\min} = 0.70$ already preserves sufficient sequence resolution for reliable selection in this setting. Performance is likewise stable for $C_{\max}$ between 128 and 512 tokens, with scores ranging from 92.16% to 92.21%. Allowing substantially larger regions ($C_{\max} = 1024$) lowers accuracy to 91.42%, indicating that excessive aggregation can make region summaries too coarse for reliable sparse selection.

Together, these results support the intended behavior of the two adaptive mechanisms: overly aggressive feature truncation or sequence aggregation can remove useful selection detail, while HASA remains robust within a broad intermediate operating range.