BEYOND LINEAR SURROGATES: SOFTMAX-AWARE INFORMATION CAPACITY FOR KV CACHE EVICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Key-value (KV) cache eviction is essential for scaling long-context inference in Large Language Models (LLMs). However, existing policies predominantly rely on empirical heuristics, lacking a principled characterization of token utility under the inherently nonlinear softmax attention mechanism. In this work, we revisit KV cache eviction through the lens of local information geometry, modeling attention as a nonlinear Gaussian communication channel. By locally linearizing the attention mapping through a first-order Taylor expansion, we derive the *Jacobian Information Capacity*, an information-theoretic objective that jointly captures query relevance, softmax sensitivity, and structural diversity. Guided by this formulation, we introduce JACAP, a capacity-aware eviction method that combines softmax-aware importance weighting with statistical leverage scores for subset selection. Extensive experiments across diverse model architectures and benchmarks show that JACAP strong and consistent improvements over existing eviction methods across most settings, particularly under high compression ratios.

1 INTRODUCTION

Large language models (LLMs) are increasingly deployed in long-horizon applications such as multi-step reasoning (Plaat et al., 2025), tool-augmented agents (Ferrag et al., 2025), and interactive decision-making systems (Zhao et al., 2025). These settings often require processing contexts spanning thousands to millions of tokens, making efficient memory management a major challenge for long-context inference. In autoregressive decoding, the key-value (KV) cache stores past attention states to avoid recomputation, but its memory footprint and access cost grow with sequence length (Shi et al., 2024a; Recasens et al., 2025). This has motivated various KV cache compression techniques (Li et al., 2024a; Shi et al., 2024b), among which cache eviction is particularly attractive for its model-agnostic nature and compatibility with existing inference pipelines.

Most KV cache eviction methods assign an importance score to each cached token and retain a subset under a fixed memory budget. Existing criteria are typically based on attention statistics (Li et al., 2024b), token recency (Xiao et al., 2023), or value magnitudes (Tian et al., 2025). Although effective empirically, such criteria provide only an indirect characterization of how cached tokens contribute to future attention computation. In particular, token-wise importance scores often neglect interactions among cache entries: two individually important tokens may carry highly redundant information, whereas a moderately scored token may be valuable because it complements the retained subset. This limitation becomes increasingly pronounced under aggressive compression, where subset structure can be as important as individual token relevance.

Recent work (Yang et al., 2026) provides a more principled perspective by formulating KV cache eviction through the Information Bottleneck principle (Tishby et al., 2000). Under a linear-Gaussian surrogate of attention, it derives an information-capacity objective that captures query relevance and representation diversity. However, real attention is inherently nonlinear due to softmax normalization and competition: a token's contribution depends not only on its query-key compatibility, but also on the responses of all competing entries. Such context-dependent sensitivity cannot be fully captured by a fixed linear transformation.

In this work, we revisit KV cache eviction from a nonlinear information-theoretic perspective. We model attention as a nonlinear communication channel and locally linearize its mapping using a first-order Taylor expansion. This yields the *Jacobian Information Capacity*, which characterizes

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.

query relevance, softmax sensitivity, and redundancy within a unified local objective. The previous linear-capacity formulation naturally arises as a special case when the attention mapping is linear.

Based on this formulation, we propose JACAP, a practical capacity-aware eviction method that combines softmax-aware importance weighting with statistical leverage scores for subset selection. Experiments across multiple model architectures and long-context benchmarks show that JACAP consistently improves over existing eviction methods in most settings, with particularly pronounced gains under aggressive cache compression.

Our contributions are summarized as follows:

1. We formulate KV cache eviction under nonlinear softmax attention as a local information-preservation problem and derive the *Jacobian Information Capacity*.
2. We introduce JACAP, a practical eviction method that combines softmax-aware weighting with leverage-score-based subset selection.
3. We evaluate JACAP across multiple LLM architectures and benchmarks, including LongBench, NIAH, and AIME25, demonstrating consistent improvements over prior methods, especially at high compression ratios.

2 RELATED WORKS

2.1 EFFICIENT LLM INFERENCE

The rapidly growing computational cost of LLM inference has motivated optimizations across multiple levels of the inference stack. At the representation level, quantization methods such as TurboQuant Zandieh et al. (2025), PolarQuant Han et al. (2025), and KVQuant Hooper et al. (2024) reduce memory footprint by storing KV Cache in low-precision formats, though they may introduce non-negligible degradation under extreme long-context settings. At the sequence level, token reduction techniques, including token merging methods such as D2O Wan et al. (2024) and CAM Zhang et al. (2024), aim to shorten the effective context length by consolidating redundant tokens. At the system level, memory management optimizations such as PagedAttention Kwon et al. (2023) and FlashDecoding++ Hong et al. (2023) improve hardware utilization by reducing KV cache fragmentation and increasing token throughput.

Orthogonal to these approaches, KV-cache eviction reduces inference memory by selectively retaining cached tokens, which we study in this work.

2.2 KV CACHE EVICTION

Heuristic-based Eviction. KV cache eviction methods aim to alleviate the memory bottleneck by selectively pruning cached tokens during autoregressive decoding. Early approaches primarily rely on heuristic importance metrics derived from observed token behavior. Attention-based methods, such as H2O Zhang et al. (2023) and SNAPKV Li et al. (2024b), prioritize tokens that receive consistently high attention weights across decoding steps. Other approaches emphasize structural diversity: KEYDIFF Park et al. (2025) promotes geometric dissimilarity among keys to reduce redundancy, while KNORM Devoto et al. (2024) uses l_2-norms as lightweight proxies for token salience. Recent work has explored estimating token importance beyond observed attention patterns. Expected Attention Devoto et al. (2025) approximates the expected attention of future queries under a distributional assumption, enabling importance scoring without access to explicit attention matrices.

Despite their empirical effectiveness, these methods typically score tokens independently and therefore fail to capture higher-order interactions such as redundancy and complementarity among retained tokens.

Information-Theoretic Perspectives. Recent work has introduced information-theoretic formulations for KV-cache eviction based on the Information Bottleneck (IB) principle Tishby et al. (2000); Conklin et al. (2026). In particular, CAPKV Yang et al. (2026) formulates token selection as maximizing the mutual information between future queries and attention outputs. Under a linear-Gaussian approximation, the objective reduces to a tractable log-determinant form.

However, this formulation is derived from a linear query-to-output mapping and does not explicitly model the nonlinear normalization and competition induced by softmax attention. Our work extends this information-theoretic perspective to nonlinear attention by characterizing local information transmission through the Jacobian of the attention mapping, yielding a local capacity objective that captures softmax-aware token interactions.

3 METHODOLOGY

In this section, we first revisit the formulation of KV cache eviction and its information-theoretic interpretation. We then introduce a local information-capacity framework that accounts for the nonlinear sensitivity of softmax attention. Based on this formulation, we propose JACAP (**Ja**cobian-**Cap**acity Aware Eviction), a practical KV cache eviction policy.

3.1 PRELIMINARY

During autoregressive decoding, each step t attends to all previously generated tokens. Given the current hidden state h_t, the model computes

$$q_t = W_Q h_t, \qquad k_i = W_K h_i, \qquad v_i = W_V h_i, \qquad i \le t.$$

Let d_k denote the query/key dimension and d_o the output dimension. The attention output is

$$o_t = \mathrm{Attn}(q_t) = W_O \sum_{i \le t} \alpha_{t,i} v_i, \qquad \alpha_{t,i} = \frac{\exp(q_t^\top k_i / \sqrt{d_k})}{\sum_{j \le t} \exp(q_t^\top k_j / \sqrt{d_k})}.$$

Modern inference systems maintain a KV cache $\{(k_i, v_i)\}_{i=1}^t$ to avoid recomputing past keys and values. However, the memory cost of the cache grows linearly with the sequence length, making KV cache eviction necessary for long-context inference.

From a functional perspective, full attention defines a nonlinear mapping from query to output:

$$f(q) = U\alpha(q), \qquad \alpha(q) = \mathrm{softmax}(Kq/\sqrt{d_k}),$$

where $u_i = W_O v_i$, $U = [u_1, \dots, u_t] \in \mathbb{R}^{d_o \times t}$, and $K = [k_1^\top; \dots; k_t^\top] \in \mathbb{R}^{t \times d_k}$. Given a memory budget $B < t$, eviction selects a subset $\mathcal{C} \subseteq \mathcal{H}_t := \{1, \dots, t\}$ with $|\mathcal{C}| \le B$, resulting in a retained cache

$$\mathcal{Z}_\mathcal{C} = \{(k_i, v_i)\}_{i \in \mathcal{C}}.$$

For the subset $\mathcal{C}$, define

$$K_\mathcal{C} = [k_i^\top]_{i \in \mathcal{C}} \in \mathbb{R}^{|\mathcal{C}| \times d_k}, \qquad U_\mathcal{C} = [u_i]_{i \in \mathcal{C}} \in \mathbb{R}^{d_o \times |\mathcal{C}|}.$$

The compressed cache induces the restricted mapping

$$f_\mathcal{C}(q) = U_\mathcal{C} \alpha_\mathcal{C}(q), \qquad \alpha_i^\mathcal{C}(q) = \frac{\exp(q^\top k_i / \sqrt{d_k})}{\sum_{j \in \mathcal{C}} \exp(q^\top k_j / \sqrt{d_k})}.$$

Thus, KV cache eviction can be formulated as a constrained function approximation problem:

$$\min_{\mathcal{C}: |\mathcal{C}| \le B} \mathbb{E}_{q \sim P(q)} \left[\|f(q) - f_\mathcal{C}(q)\|_2^2 \right],$$

where $P(q)$ denotes the distribution of future queries.

3.2 AN INFORMATION BOTTLENECK PERSPECTIVE ON KV CACHE EVICTION

The Information Bottleneck (IB) principle provides a formal view of the trade-off between compression and predictive information (Tishby & Zaslavsky, 2015; Chechik et al., 2003). In KV cache eviction, however, representations are produced by a frozen model, while compression is imposed through a hard cache budget rather than a learned encoder. Therefore, instead of optimizing the full IB Lagrangian, we focus on the predictive information preserved by the retained cache.

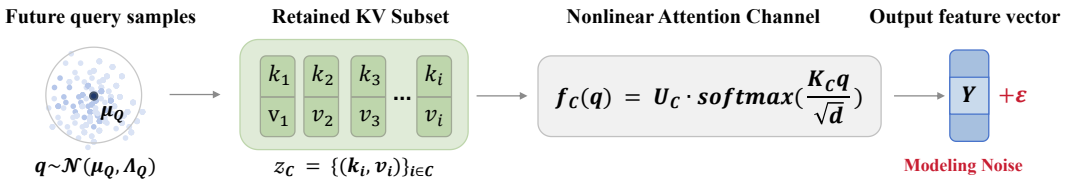


Figure 1: **Nonlinear retained attention channel.** A retained KV subset $\mathcal{Z}_\mathcal{C}$ defines a nonlinear softmax attention map from future queries $q \sim \mathcal{N}(\mu_Q, \Lambda_Q)$ to output features, $Y_\mathcal{C} = f_\mathcal{C}(q) + \epsilon$.

Given the retained cache $\mathcal{Z}_\mathcal{C}$, we model the future query q as a random variable and define the stochastic channel output as

$$Y_\mathcal{C} = f_\mathcal{C}(q) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \Sigma_{\text{noise}}), \qquad q \perp \epsilon.$$

The information preserved by the retained cache is then measured by

$$\mathcal{L}_\mathcal{C} = I(q; Y_\mathcal{C} \mid \mathcal{Z}_\mathcal{C}). \tag{1}$$

Prior work (Yang et al., 2026) instantiates this objective with a linear-gaussian surrogate,

$$Y_\mathcal{C}^{\text{Lin}} = U_\mathcal{C} K_\mathcal{C} q + \epsilon, \qquad q \sim \mathcal{N}(\mu_Q, \Lambda_Q).$$

Under this model, the capacity admits the closed form

$$\mathcal{L}_\mathcal{C}^{\text{Lin}} = \frac{1}{2} \log\det\left(I + \Sigma_{\text{noise}}^{-1} U_\mathcal{C} K_\mathcal{C} \Lambda_Q K_\mathcal{C}^\top U_\mathcal{C}^\top\right). \tag{2}$$

3.3 Jacobian Information Capacity for KV Cache Selection

The linear–Gaussian surrogate above provides a tractable objective for KV cache selection. However, the actual attention mapping induced by a retained cache is nonlinear, as softmax normalization couples cached entries and introduces competition among their attention weights. We therefore model the retained cache directly as a nonlinear query-to-output mapping.

Given a retained KV subset $\mathcal{Z}_\mathcal{C} = \{(k_i, v_i)\}_{i \in \mathcal{C}}$, we denote by $K_\mathcal{C}$ the matrix of retained keys and by $U_\mathcal{C}$ the corresponding value-induced output directions. The retained attention map is defined as

$$f_\mathcal{C}(q) = U_\mathcal{C} \,\text{softmax}\left(\frac{K_\mathcal{C} q}{\sqrt{d_k}}\right). \tag{3}$$

We then model the output representation produced by the retained cache as a noisy nonlinear channel:

$$Y_\mathcal{C} = f_\mathcal{C}(q) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \Sigma_{\text{noise}}). \tag{4}$$

As illustrated in Fig. 1, this formulation preserves the nonlinear softmax normalization that is absent from the global linear surrogate. The retained cache therefore defines a coupled nonlinear attention channel whose response depends jointly on query–key accessibility and token competition.

To obtain a tractable local capacity measure, we assume that future queries concentrate around a distribution center μ_Q:

$$q = \mu_Q + \delta q, \qquad \delta q \sim \mathcal{N}(0, \Lambda_Q).$$

Under this local query model, the nonlinear map in Eq. equation 3 can be approximated by its first-order Taylor expansion around μ_Q.

Lemma 1 (Local Jacobian Linearization). *For a retained cache $\mathcal{Z}_\mathcal{C}$, the nonlinear attention map $f_\mathcal{C}(q)$ admits the first-order expansion*

$$f_\mathcal{C}(q) = f_\mathcal{C}(\mu_Q) + J_\mathcal{C} \delta q + \mathcal{O}(\|\delta q\|_2^2), \tag{5}$$

where $J_\mathcal{C} \in \mathbb{R}^{d_o \times d_k}$ is

$$J_\mathcal{C} = \frac{1}{\sqrt{d_k}} U_\mathcal{C} S_\mathcal{C} K_\mathcal{C}, \quad S_\mathcal{C} = \text{Diag}(\alpha_\mathcal{C}^\star) - \alpha_\mathcal{C}^\star (\alpha_\mathcal{C}^\star)^\top, \quad \alpha_\mathcal{C}^\star = \text{softmax}(K_\mathcal{C} \mu_Q / \sqrt{d_k}). \tag{6}$$

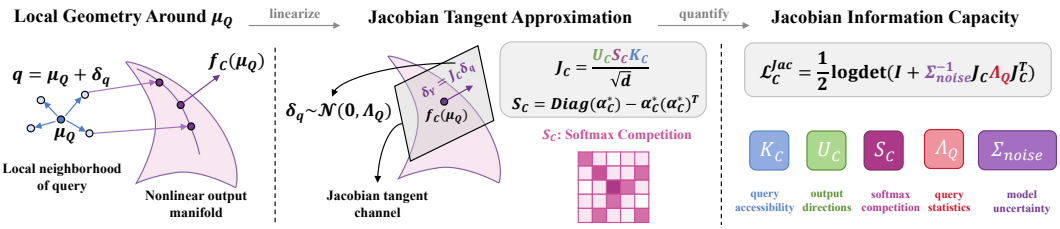


Figure 2: **Jacobian Information Capacity.** Around the center μ_Q of the future query distribution, the nonlinear retained attention map is locally approximated by its Jacobian tangent channel. The Jacobian $J_\mathcal{C} = U_\mathcal{C} S_\mathcal{C} K_\mathcal{C} / \sqrt{d_k}$ decomposes the local channel into key accessibility $K_\mathcal{C}$, output directions $U_\mathcal{C}$, and the softmax competition matrix $S_\mathcal{C}$. This yields a local information-capacity objective $\mathcal{L}_\mathcal{C}^{\text{Jac}}$, which quantifies how much query-dependent information is preserved by the retained cache under query statistics Λ_Q and modeling uncertainty Σ_{noise}.

Treating $J_\mathcal{C}$ as the effective local channel gain yields the Jacobian-capacity objective.

Theorem 1 (Jacobian Information Capacity)**.** *Under the local Gaussian perturbation model above and the first-order approximation in Lemma 1, the retained cache capacity is approximated by*

$$\begin{aligned}\mathcal{L}_\mathcal{C}^{\text{Jac}} &\approx \frac{1}{2} \log\det\left(I + \Sigma_{\text{noise}}^{-1} J_\mathcal{C} \Lambda_Q J_\mathcal{C}^\top\right) \\ &= \frac{1}{2} \log\det\left(I + \frac{1}{d_k} \Sigma_{\text{noise}}^{-1} U_\mathcal{C} S_\mathcal{C} K_\mathcal{C} \Lambda_Q K_\mathcal{C}^\top S_\mathcal{C} U_\mathcal{C}^\top\right).\end{aligned} \tag{7}$$

The key difference from the linear capacity in Eq. equation 2 lies in the softmax Jacobian $S_\mathcal{C}$. Its diagonal entries measure the local sensitivity of each token's attention probability to its own logit, while its off-diagonal entries encode the normalization-induced competition between different tokens. Consequently, $\mathcal{L}_\mathcal{C}^{\text{Jac}}$ favors retained subsets whose output directions remain informative under the local nonlinear response, rather than considering query alignment alone. As illustrated in Fig. 2, the resulting capacity decomposes into five interpretable components: key accessibility, output directions, softmax competition, query statistics, and model uncertainty.

Remark 1. *Under standard smoothness and non-degenerate noise assumptions, the Jacobian approximation matches the first-order local channel induced by nonlinear attention. The remaining approximation error is determined by higher-order Taylor residuals. We provide the detailed derivation and analysis of Theorem 1 in Appendix A.*

3.4 JACAP: A JACOBIAN-CAPACITY AWARE EVICTION POLICY

Although Theorem 1 provides a principled objective for measuring cache utility, directly optimizing the subset-dependent log-determinant objective during online inference is computationally expensive. Our goal is therefore not to optimize the full Jacobian capacity exactly, but to preserve its essential structure in a lightweight eviction score. Following the decomposition in Fig. 2, we retain three ingredients that are most relevant for token selection: local query relevance, softmax sensitivity, and output-space diversity.

The first difficulty comes from the subset-dependent local attention distribution $\alpha_\mathcal{C}^\star$. Since the retained subset $\mathcal{C}$ is unknown before eviction, we compute the local attention distribution over the pre-eviction candidate pool $\mathcal{H}_t$:

$$\bar{\alpha}_i = \frac{\exp\left(k_i^\top \mu_Q / (\tau\sqrt{d_k})\right)}{\sum_{j \in \mathcal{H}_t} \exp\left(k_j^\top \mu_Q / (\tau\sqrt{d_k})\right)}, \qquad i \in \mathcal{H}_t.$$

Here τ denotes the temperature. We use $\bar{\alpha}_i$ as a subset-independent approximation to the local attention weight of token i.

The second difficulty comes from the dense softmax competition matrix $S_\mathcal{C}$, which couples all tokens in the retained subset. For efficient scoring, we retain its diagonal sensitivity and approximate

$$S_\mathcal{C} \approx \text{Diag}(\rho_i), \qquad \rho_i = \bar{\alpha}_i(1 - \bar{\alpha}_i).$$

This approximation keeps the local sensitivity induced by softmax saturation while avoiding pairwise token coupling. Under this approximation, the local Jacobian reduces to a weighted key–output interaction.

We further approximate the query-response covariance $K\Lambda_Q K^\top$ by its diagonal entries

$$\kappa_i = k_i^\top \Lambda_Q k_i,$$

which measure the variance of the i-th key response under future query perturbations. Combining the diagonal softmax sensitivity and the diagonal query covariance yields the nonlinear importance weight

$$w_i = \frac{1}{d_k}\rho_i^2 \kappa_i = \frac{1}{d_k}\bar{\alpha}_i^2(1-\bar{\alpha}_i)^2\kappa_i. \tag{8}$$

The weight w_i accounts for three local factors: query relevance through $\bar{\alpha}_i$, softmax sensitivity through $\bar{\alpha}_i(1-\bar{\alpha}_i)$, and query-space variability through κ_i. This form differs from pure attention-based scoring: tokens with extremely large attention may become saturated and thus contribute limited local sensitivity, while tokens with moderate but responsive attention can carry larger Jacobian capacity.

With these approximations, the Jacobian capacity reduces to the log-determinant of a weighted output-space capacity matrix. For one-shot selection, we construct this matrix over the full candidate pool:

$$A_{\mathrm{Jac}} = I + \sum_{j \in \mathcal{H}_t} w_j u_j u_j^\top. \tag{9}$$

This matrix summarizes the output directions already represented by the candidate cache. Applying the matrix determinant lemma motivates the following token score:

$$r_i^{\mathrm{Jac}} = w_i \, u_i^\top A_{\mathrm{Jac}}^{-1} u_i. \tag{10}$$

Algorithm 1: JACAP: Jacobian-Capacity-Aware KV Eviction

Require : KV cache $\{(k_i, v_i)\}_{i=1}^N$, cache budget B, query statistics (μ_Q, Λ_Q), temperature τ
Output : Updated retained KV subset $\mathcal{C}$ with $|\mathcal{C}| = B$

$\alpha_i^\star \leftarrow \mathrm{softmax}\big(K\mu_Q/(\tau\sqrt{d_k})\big)_i$; `// attention score at query center`
$s_i \leftarrow \alpha_i^{\star 2}(1-\alpha_i^\star)^2 \cdot (k_i^\top \Lambda_Q k_i)$; `// softmax-sensitive weight`
$A \leftarrow I + \sum_{i=1}^N s_i \, u_i u_i^\top$; `// jacobian capacity matrix`
$\mathrm{score}_i \leftarrow s_i \, u_i^\top A^{-1} u_i$; `// capacity-aware token score`
$\mathcal{C} \leftarrow$ TOP-B indices according to score_i
return $\mathcal{C}$

In practice, μ_Q and Λ_Q are estimated from historical queries observed during decoding. When applying the output projection is undesirable, we use $u_i = v_i$ as a lightweight proxy for $W_O v_i$, following prior capacity-based eviction methods. The full procedure is summarized in Algorithm 1. We provide the detailed derivation of JACAP together with its computational complexity analysis in Appendix B. We further conduct empirical runtime evaluations, which show that JACAP achieves comparable efficiency to existing mainstream methods in practice. Detailed runtime comparisons are reported in Appendix C.1.

4 EXPERIMENT

We evaluate our proposed JACAP eviction strategy against representative baselines, including CAPKV Yang et al. (2026), SNAPKV Li et al. (2024b), KEYDIFF Park et al. (2025), Expected Attention (EA) Devoto et al. (2025), and KNORM Devoto et al. (2024). The experiments are designed to assess the effectiveness of KV-cache compression in preserving predictive information, handling extremely long contexts, and supporting online decoding under dynamic eviction.

Table 1: Performance of JACAP and baseline KV-cache eviction methods on LongBench across different compression ratios (C.R.). For each compression ratio, **bold** and underlined denote the best and second-best results, respectively.

		SD-QA			MD-QA			SUMM.			FSL			SYNTH.		CODE		
	C.R.	NQA	QSER	MF	HPQA	2WQA	MQUE	GR	QMS	MN	TREC	TQA	SSUM	PC	PR	LCC	RBP	AVG.
QWEN3-8B		28.89	43.57	54.76	62.86	49.20	35.69	33.67	24.60	24.99	41.00	90.21	39.95	10.50	91.02	66.83	61.95	47.48
CAPKV	0.25	28.30	**44.13**	**54.26**	**62.95**	**48.47**	**34.93**	33.55	24.21	**25.05**	**72.00**	88.47	41.00	10.50	**95.54**	65.60	**63.23**	**49.51**
	0.5	**29.43**	41.84	**51.11**	61.15	44.40	**34.08**	33.09	24.25	24.59	66.50	87.49	39.64	**11.00**	**98.42**	60.06	**63.32**	48.15
	0.75	**28.62**	37.69	**44.16**	54.33	38.44	**30.54**	30.99	**23.58**	23.12	60.50	**85.67**	38.29	9.50	**96.96**	51.92	**63.81**	44.88
	0.9	**24.60**	27.49	**36.12**	47.69	31.49	25.12	27.11	21.61	20.27	36.00	85.97	35.00	6.50	57.38	39.65	**63.06**	36.57
EA	0.25	**28.69**	42.84	53.02	62.23	48.37	33.99	33.61	**24.22**	24.94	57.00	88.13	40.25	8.00	91.81	**66.33**	62.17	47.85
	0.5	28.34	39.54	**51.11**	61.22	**46.78**	30.13	33.38	24.00	**24.94**	64.00	86.96	39.81	8.50	87.46	**65.41**	62.30	47.12
	0.75	27.23	34.65	39.60	57.52	39.27	27.58	32.04	22.76	23.70	70.00	85.29	**39.71**	**10.00**	49.74	**59.30**	63.46	42.62
	0.9	23.28	27.76	32.42	43.62	31.74	23.28	29.65	20.98	22.00	61.75	84.16	**38.15**	**11.50**	17.42	**50.93**	62.92	36.35
KEYDIFF	0.25	25.44	35.02	49.12	47.28	42.49	23.07	32.84	22.94	24.93	62.50	83.61	38.84	6.37	91.50	61.75	56.71	44.03
	0.5	23.30	29.97	39.86	35.86	33.86	16.60	29.49	22.43	21.93	55.00	85.62	38.25	5.49	86.33	46.51	53.65	39.01
	0.75	14.90	18.21	29.26	19.68	27.49	9.51	21.97	21.18	14.22	39.00	81.56	36.15	5.81	45.58	27.38	53.66	29.10
	0.9	10.62	7.80	23.18	15.76	23.01	7.01	16.07	19.85	8.02	6.50	78.15	31.41	7.78	13.92	13.57	52.57	20.95
KNORM	0.25	25.35	34.04	49.59	49.24	40.66	24.62	32.53	23.17	24.51	63.00	84.55	**41.94**	11.50	93.00	50.21	47.71	43.48
	0.5	17.84	24.04	39.55	29.63	28.61	14.82	28.85	21.55	21.26	50.50	81.30	**40.80**	6.64	78.46	36.13	51.59	35.72
	0.75	13.39	12.33	27.76	13.90	18.25	6.39	22.63	20.24	14.73	25.75	81.36	38.51	9.88	20.00	16.83	54.66	24.79
	0.9	9.66	7.65	23.48	8.78	19.54	4.11	15.46	19.24	8.15	14.00	80.68	33.08	3.75	6.50	11.54	55.83	20.09
JACAP (OURS)	0.25	28.61	43.45	53.05	62.37	48.04	33.94	**33.84**	24.09	24.72	71.50	**90.01**	40.63	**12.50**	**95.54**	65.10	63.02	49.40
	0.5	28.40	**43.92**	50.90	**62.61**	46.25	34.04	**33.48**	**24.47**	24.87	**75.00**	**87.82**	40.01	9.50	96.71	62.24	62.91	**48.95**
	0.75	27.70	**39.07**	43.48	**59.54**	**42.13**	28.50	**32.60**	23.44	**23.89**	**72.50**	84.36	39.55	**10.00**	95.96	57.65	63.36	**46.48**
	0.9	24.12	**31.39**	34.32	**49.68**	**37.26**	**28.34**	**30.21**	**22.80**	**22.14**	**62.25**	**86.90**	36.56	9.00	**70.29**	46.93	62.37	**40.91**
QWEN3-14B		30.32	44.05	52.14	62.30	55.87	32.79	32.84	24.32	24.96	70.00	89.10	42.01	9.80	99.42	69.36	67.27	50.41
CAPKV	0.25	**30.56**	43.25	**52.49**	**63.55**	54.25	**35.68**	32.85	**24.66**	24.88	**75.00**	90.10	41.80	8.00	99.08	**69.42**	**67.25**	**50.80**
	0.5	**29.92**	**43.96**	**51.95**	**65.06**	54.11	**35.42**	32.80	24.19	24.87	72.00	**91.85**	41.27	9.50	**97.62**	66.82	**67.84**	**50.57**
	0.75	27.62	37.53	45.64	**58.93**	46.67	32.22	31.09	24.24	23.70	66.00	89.71	40.80	7.15	**98.58**	57.86	66.89	47.16
	0.9	25.16	27.78	35.16	49.75	32.62	26.88	27.45	21.98	20.70	48.25	87.82	38.41	5.50	76.67	45.08	63.25	39.53
EA	0.25	29.31	**44.25**	51.06	63.10	**56.44**	32.64	32.79	24.47	24.85	70.50	89.60	41.49	9.00	97.45	69.30	66.54	50.17
	0.5	28.28	43.70	49.47	63.83	50.77	31.47	32.29	24.13	25.15	70.50	90.56	41.46	8.70	94.17	**67.75**	65.82	49.25
	0.75	23.62	35.92	37.70	52.92	38.91	26.90	31.37	22.79	24.74	66.50	**90.10**	**41.20**	8.55	72.74	61.32	64.21	43.72
	0.9	22.78	25.16	27.36	41.20	23.29	20.64	28.79	21.28	22.39	61.42	**89.02**	**39.65**	**10.50**	27.96	50.87	62.00	35.89
KEYDIFF	0.25	28.11	36.18	50.24	51.61	45.09	26.48	33.02	24.06	24.94	69.00	82.57	41.52	**11.50**	98.58	66.02	62.22	46.95
	0.5	24.93	30.20	44.89	41.04	35.61	21.60	28.69	23.16	22.65	57.50	75.73	39.56	**13.00**	95.50	49.69	57.42	41.32
	0.75	21.03	21.80	34.10	26.79	27.78	14.63	20.31	21.41	15.55	33.00	74.33	34.99	10.50	81.83	28.68	54.69	32.59
	0.9	12.76	12.12	24.13	18.97	25.03	6.77	11.41	19.50	6.68	3.00	74.32	30.10	**10.50**	39.00	15.19	54.80	22.77
KNORM	0.25	28.28	40.58	52.17	54.61	43.52	28.23	**33.24**	23.74	24.95	65.00	79.56	39.75	10.50	**100.00**	58.96	62.29	46.59
	0.5	23.58	34.56	44.65	43.17	36.84	15.95	26.18	22.96	22.80	60.00	78.35	39.15	12.50	95.08	44.16	59.78	41.23
	0.75	17.72	17.80	35.08	22.71	25.99	6.97	15.05	20.99	16.87	44.50	82.07	36.27	**13.00**	63.58	27.58	54.15	31.27
	0.9	10.07	16.59	26.61	11.10	21.02	4.54	10.45	19.66	11.37	23.50	77.52	33.51	8.50	13.58	20.85	46.55	22.21
JACAP (OURS)	0.25	29.28	43.70	52.14	62.43	56.03	35.50	32.96	24.50	**25.13**	71.00	**90.85**	**41.95**	9.00	97.75	69.24	67.20	50.54
	0.5	28.50	42.62	51.19	63.26	**54.65**	34.06	**32.81**	**24.79**	**25.22**	**74.00**	90.31	**42.27**	7.05	**97.62**	67.43	67.32	50.19
	0.75	**27.90**	**41.32**	**45.78**	57.85	**52.86**	**33.29**	**32.09**	**24.83**	**24.86**	**75.50**	88.94	41.08	7.99	96.92	**62.58**	**67.92**	**48.86**
	0.9	**27.19**	**32.42**	**38.16**	**54.21**	**45.41**	**30.60**	**29.59**	**22.62**	**23.21**	**68.50**	88.57	38.62	8.56	**88.08**	**51.73**	**66.00**	**44.59**
LLAMA3.1-8B		30.46	47.27	55.98	59.00	51.23	33.55	35.24	25.27	26.80	29.50	86.01	39.31	10.65	100.00	53.47	47.73	45.72
CAPKV	0.25	**31.53**	**47.74**	**56.95**	57.31	**51.90**	32.91	34.61	24.65	26.78	**68.50**	**90.21**	38.80	**13.65**	**100.00**	53.47	48.52	**48.60**
	0.5	31.69	44.84	51.42	**58.24**	48.35	30.66	**33.88**	24.65	26.47	65.25	**90.71**	37.72	12.00	98.50	50.12	49.47	47.12
	0.75	29.73	39.93	43.71	53.19	**46.49**	24.66	29.32	23.71	25.19	30.75	90.83	33.44	9.00	66.50	44.47	**50.91**	40.11
	0.9	25.33	24.72	27.81	43.90	30.86	21.58	24.56	22.14	21.54	7.00	91.22	28.79	10.00	21.50	35.61	51.06	30.48
EA	0.25	31.17	47.49	56.85	58.33	49.98	33.82	34.94	24.78	27.18	29.50	85.66	38.03	11.20	99.00	52.49	47.54	45.50
	0.5	31.19	46.67	51.49	56.34	48.65	**33.48**	33.63	24.74	27.04	19.00	85.91	38.87	**12.50**	87.00	**52.97**	48.64	43.63
	0.75	30.17	39.70	41.55	**55.53**	44.55	26.75	**31.77**	23.26	**26.06**	27.00	88.98	38.59	**10.00**	45.00	**49.84**	50.01	39.30
	0.9	25.87	29.04	32.51	48.46	31.27	24.48	**29.07**	**22.72**	**24.60**	**50.50**	**92.17**	26.90	7.55	18.00	40.11	48.94	34.51
KEYDIFF	0.25	31.50	47.20	54.17	**58.79**	50.95	**35.85**	34.48	24.77	26.83	62.50	86.56	**40.13**	10.65	99.50	52.25	47.03	47.70
	0.5	**32.30**	44.93	**52.98**	55.52	46.60	30.26	32.45	24.47	25.71	64.00	87.29	40.35	10.65	**99.50**	45.18	46.77	46.18
	0.75	29.62	33.01	41.37	52.51	37.97	25.28	29.41	**23.95**	22.93	52.00	84.36	**40.03**	9.66	**99.50**	37.48	47.58	41.67
	0.9	28.49	16.95	34.22	40.71	25.38	15.27	26.30	21.03	19.15	40.00	80.01	**39.18**	8.50	**89.00**	26.07	47.99	34.89
KNORM	0.25	30.71	45.34	56.16	58.43	50.63	35.27	34.07	24.83	26.67	66.00	87.26	36.85	11.15	97.50	36.91	48.35	46.63
	0.5	27.45	39.59	49.46	55.72	44.23	26.95	32.22	24.25	25.51	59.00	86.44	33.60	9.55	87.50	33.05	49.16	42.73
	0.75	22.18	29.67	39.44	47.45	30.45	20.51	28.68	22.78	23.48	51.50	72.85	28.98	9.15	52.50	27.20	49.17	34.75
	0.9	22.44	16.23	28.48	37.69	21.73	13.30	26.03	20.64	20.56	45.50	62.78	29.03	**11.50**	25.00	23.06	**51.52**	28.47
JACAP (OURS)	0.25	31.46	47.52	55.57	57.79	50.26	32.54	**35.04**	**24.99**	**27.30**	44.50	90.06	39.82	11.15	**100.00**	**53.68**	**50.50**	47.01
	0.5	30.32	**46.74**	52.37	55.51	**48.92**	27.33	33.71	**24.75**	**27.14**	**66.00**	89.31	**40.59**	11.00	99.00	51.78	**49.87**	**47.15**
	0.75	**30.77**	**42.32**	**45.32**	54.92	45.89	**28.65**	31.05	23.34	25.94	**56.50**	**91.96**	37.26	7.00	87.50	48.04	50.51	**44.19**
	0.9	**29.13**	**30.94**	**35.46**	**49.26**	**34.00**	**25.15**	27.52	22.64	24.48	32.50	90.40	27.93	8.00	46.50	**40.52**	51.21	**35.98**

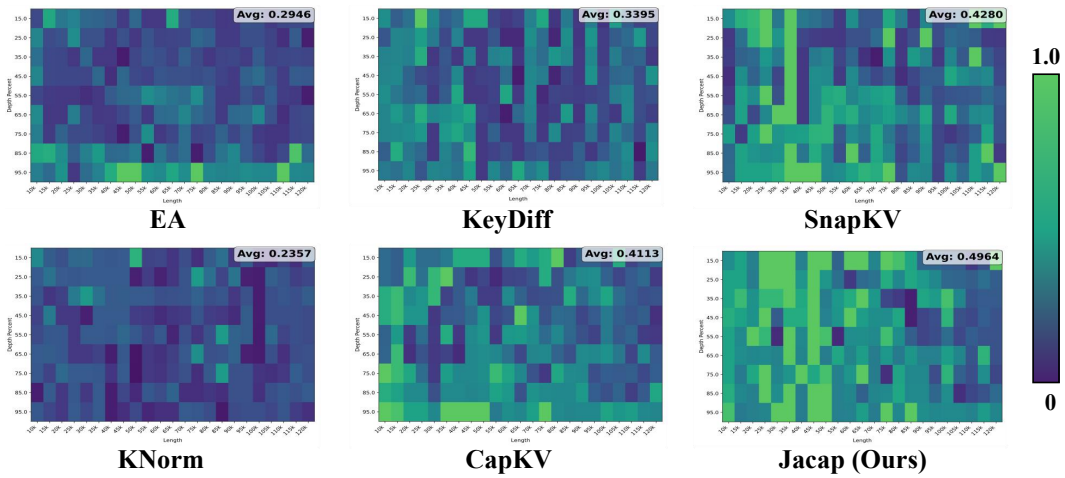


Figure 3: Performance of JACAP and baseline KV-cache eviction methods on the Needle-in-Haystack (NIAH) benchmark at compression ratio 0.75 using Qwen3-8B.

4.1 EXPERIMENTS ON LONGBENCH

We evaluate JACAP against CAPKV, KEYDIFF, EA, and KNORM on the LongBench Bai et al. (2023) across four compression ratios (0.25, 0.5, 0.75, 0.9), using Qwen3-8B, Qwen3-14B Yang et al. (2025), and Llama3.1-8B Grattafiori et al. (2024). Table 1 summarizes the results. We focus our discussion on high compression ratios (0.75 and 0.9), where cache budgets are most constrained and performance differences among methods become more pronounced.

At 0.75 compression, JACAP generally matches or slightly exceeds CAPKV on average, while significantly outperforming heuristic baselines (KEYDIFF, EA, KNORM). When the compression ratio increases to 0.9, the advantage becomes larger: averaged across the three models, JACAP exceeds the strongest baseline by approximately 5 points. This demonstrates its superior ability to retain critical KV information under extreme cache constraints.

At the task level, JACAP shows strong performance on both SD-QA and MD-QA tasks, particularly at high compression ratios, suggesting better preservation of information relevant to long-context reasoning. It also exhibits more stable degradation on summarization and few-shot tasks as compression increases. These trends remain consistent across all three models, demonstrating the robustness and generality of the proposed method.

4.2 EXPERIMENTS ON NIAH

We evaluate the performance of JACAP and baseline KV-cache eviction methods on the Needle-in-Haystack (NIAH) Kamradt (2023) benchmark under a compression ratio of 0.75 using Qwen3-8B. The NIAH setup tests extremely long contexts (10k–120k tokens) and varying Needle Depths (0.15–0.95), simulating sparse target retrieval under constrained cache budgets.

Figure 3 presents the retrieval performance heatmaps across different context lengths and Needle Depths. JACAP consistently achieves stronger retrieval performance, particularly for deep needles and long contexts where heuristic baselines show significant degradation. Although CAPKV performs competitively at moderate depths, JACAP maintains more stable performance across the full range of settings. These results suggest that Jacobian-capacity-aware eviction more effectively preserves KV pairs relevant to sparse and distant queries, leading to more reliable retrieval in extreme long-context scenarios. Additional results are provided in Appendix C.2.

4.3 EXPERIMENTS ON DECODING EVICTION

We evaluate JACAP during online decoding on AIME25 Zhang & Math-AI (2025) using Nemotron-7B Ahmad et al. (2025). Table 2 summarizes performance across different reserved token budgets, compared with EA, KEYDIFF, KNORM, and CAPKV. At the smallest cache budget (2048 tokens),

Table 2: Performance comparison of different KV-cache eviction methods on AIME25.

Reserved Tokens	EA	KeyDiff	KNorm	CapKV	Jacap (Ours)
2048	0.17	0.07	0	0.20	**0.30**
4096	0.30	0.33	0.03	0.50	0.50
8192	0.40	0.53	0.27	**0.73**	0.53
16384	0.63	0.67	0.57	0.67	**0.73**

JACAP achieves the best score, showing its effectiveness in preserving critical KV entries under extreme constraints. With 4096 reserved tokens, it matches CAPKV and clearly outperforms other baselines. At 8192 tokens, CAPKV surpasses JACAP, but when the budget increases to 16384 tokens, JACAP recovers the lead, outperforming all methods.

Overall, JACAP remains competitive under dynamic decoding, achieving the best or tied-best performance at three of the four evaluated cache budgets. Its sensitivity-aware, non-redundant selection helps maintain reasoning performance throughout autoregressive generation, highlighting the practical benefits of Jacobian-capacity-aware eviction.

4.4 ABLATION STUDY

To evaluate the diagonal approximation and isolate the contributions of individual components, we compare five variants of JACAP on LongBench using Qwen3–8B: (1) **Full Jacobian**, (2) **JACAP** with the diagonal approximation, (3) **w/o Sensitivity** ($\rho_i \to 1$), (4) **w/o Leverage**, and (5) **w/o Query Covariance** ($\kappa_i \to 1$). The results are summarized in Table 3. Additional ablation studies concerning the temperature parameter τ are provided in Appendix C.3.

Table 3: Ablation study of different components in JACAP on LongBench. We report the average score under different compression ratios.

C.R.	JACAP	Full Jacobian	w/o Sensitivity	w/o Leverage	w/o Query Covariance
0.25	49.36	49.32	49.44	48.11	48.69
0.50	48.92	49.31	48.04	46.00	47.10
0.75	46.61	47.16	41.77	39.77	41.00
0.90	40.69	42.36	33.67	32.15	33.59

The full Jacobian generally performs best under high compression, indicating that explicit token competition provides additional modeling capacity. However, it requires dense pairwise interactions with $O(N^2)$ overhead. The diagonal approximation used by JACAP, $\rho_i = \bar{\alpha}_i(1 - \bar{\alpha}_i)$, removes this coupling while retaining most of the performance gains, yielding a substantially more efficient approximation.

Removing any individual component degrades performance, with the effect becoming especially pronounced at high compression ratios. These results show that nonlinear sensitivity, query uncertainty, and output-space diversity provide complementary contributions to JACAP.

5 CONCLUSION

In this work, we introduced a local nonlinear information-theoretic framework for KV cache eviction that captures the competition and sensitivity induced by softmax attention, and proposed JACAP, a Jacobian-capacity-aware eviction method combining softmax-sensitive importance weighting with diversity-aware subset selection. Experiments across long-context benchmarks, extreme-context retrieval, and dynamic decoding show that JACAP improves over heuristic and capacity-aware baselines in most settings, with particularly strong gains under aggressive compression. Our formulation relies on a first-order local approximation and a diagonal approximation of softmax competition for efficient one-shot selection, which may become less accurate under substantial query-distribution shifts or stronger higher-order interactions. Future work could explore more expressive local approximations and adaptive cache management across layers and decoding stages.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and polishing of the manuscript, including improving grammar, clarity, conciseness, and academic style. The research ideas, methodological development, theoretical derivations, experimental design, implementation, and interpretation of results were conducted by the authors. All AI-assisted text was carefully reviewed and revised by the authors, who take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

Our implementation is built upon the KVPress framework Devoto et al. (2025), and we provide the implementation of the proposed JACAP press in the supplementary materials. The released code follows the standard KVPress interface and is designed to be integrated into its existing inference and evaluation pipeline. Readers may refer to the official KVPress documentation for framework installation, dependency setup, and general usage instructions. The experimental environment, including the operating system, hardware configuration, is documented in Appendix C. Together with the released JACAP implementation and the standard KVPress evaluation pipeline, these materials provide implementation and experimental details intended to facilitate reproduction.

To support verification of the methodological contributions, we provide detailed derivations of the Jacobian Information Capacity and the approximations underlying JACAP in the Appendix A, including the local Jacobian linearization, the resulting information-capacity objective, and the approximations used to obtain the practical eviction score. We also include the computational complexity analysis of JACAP to clarify its implementation and runtime characteristics.

REFERENCES

Wasi Uddin Ahmad, Sean Narenthiran, Somshubra Majumdar, Aleksander Ficek, Siddhartha Jain, Jocelyn Huang, Vahid Noroozi, and Boris Ginsburg. Opencodereasoning: Advancing data distillation for competitive coding. *CoRR*, abs/2504.01943, 2025. doi: 10.48550/ARXIV.2504.01943. URL https://doi.org/10.48550/arXiv.2504.01943.

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, 2023.

Gal Chechik, Amir Globerson, Naftali Tishby, and Yair Weiss. Information bottleneck for gaussian variables. *Advances in Neural Information Processing Systems*, 16, 2003.

Henry C. Conklin, Tom Hosking, Tan Yi-Chern, Julian Gold, Jonathan D. Cohen, Thomas L. Griffiths, Max Bartolo, and Seraphina Goldfarb-Tarrant. Learning is forgetting: Llm training as lossy compression, 2026. URL https://arxiv.org/abs/2604.07569.

Alessio Devoto, Yu Zhao, Simone Scardapane, and Pasquale Minervini. A simple and effective l_2 norm-based strategy for kv cache compression. *arXiv preprint arXiv:2406.11430*, 2024.

Alessio Devoto, Maximilian Jeblick, and Simon Jégou. Expected attention: Kv cache compression by estimating attention from future queries distribution. *arXiv preprint arXiv:2510.00636*, 2025. URL https://arxiv.org/abs/2510.00636.

Mohamed Amine Ferrag, Norbert Tihanyi, and Merouane Debbah. From llm reasoning to autonomous ai agents: A comprehensive review. *arXiv preprint arXiv:2504.19678*, 2025.

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.

Insu Han, Praneeth Kacham, Amin Karbasi, Vahab Mirrokni, and Amir Zandieh. Polarquant: Quantizing kv caches with polar transformation. *arXiv preprint arXiv:2502.02617*, 2025.

Ke Hong, Guohao Dai, Jiaming Xu, Qiuli Mao, Xiuhong Li, Jun Liu, Kangdi Chen, Yuhan Dong, and Yu Wang. Flashdecoding++: Faster large language model inference on gpus. *arXiv preprint arXiv:2311.01282*, 2023.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun S Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *Advances in Neural Information Processing Systems*, 37:1270–1303, 2024.

Greg Kamradt. Needle in a haystack - pressure testing llms. https://github.com/gkamradt/LLMTest_NeedleInAHaystack, 2023.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph 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, 2023.

Haoyang Li, Yiming Li, Anxin Tian, Tianhao Tang, Zhanchao Xu, Xuejia Chen, Nicole Hu, Wei Dong, Qing Li, and Lei Chen. A survey on large language model acceleration based on kv cache management. *arXiv preprint arXiv:2412.19442*, 2024a.

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. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024b.

Junyoung Park, Dalton Jones, Matthew J Morse, Raghavv Goel, Mingu Lee, and Chris Lott. Keydiff: Key similarity-based kv cache eviction for long-context llm inference in resource-constrained environments. *arXiv preprint arXiv:2504.15364*, 2025.

Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki Van Stein, and Thomas Bäck. Multi-step reasoning with large language models, a survey. *ACM Computing Surveys*, 58(6):1–35, 2025.

Pol G Recasens, Ferran Agullo, Yue Zhu, Chen Wang, Eun Kyung Lee, Olivier Tardieu, Jordi Torres, and Josep Ll Berral. Mind the memory gap: Unveiling gpu bottlenecks in large-batch llm inference. *arXiv preprint arXiv:2503.08311*, 2025.

Luohe Shi, Hongyi Zhang, Yao Yao, Zuchao Li, and Hai Zhao. Keep the cost down: A review on methods to optimize llm' s kv-cache consumption. *CoRR*, abs/2407.18003, 2024a. doi: 10.48550/ARXIV.2407.18003. URL https://doi.org/10.48550/arXiv.2407.18003.

Luohe Shi, Hongyi Zhang, Yao Yao, Zuchao Li, and Hai Zhao. Keep the cost down: A review on methods to optimize llm's kv-cache consumption. *arXiv preprint arXiv:2407.18003*, 2024b.

Yuxuan Tian, Zihan Wang, Yebo Peng, Aomufei Yuan, Zhiming Wang, Bairen Yi, Xin Liu, Yong Cui, and Tong Yang. Keepkv: Eliminating output perturbation in kv cache compression for efficient llms inference. *arXiv preprint arXiv:2504.09936*, 2025.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop, ITW 2015, Jerusalem, Israel, April 26 - May 1, 2015*, pp. 1–5. IEEE, 2015. doi: 10.1109/ITW.2015.7133169. URL https://doi.org/10.1109/ITW.2015.7133169.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Zhongwei Wan, Xinjian Wu, Yu Zhang, Yi Xin, Chaofan Tao, Zhihong Zhu, Xin Wang, Siqi Luo, Jing Xiong, Longyue Wang, et al. D2o: Dynamic discriminative operations for efficient long-context inference of large language models. *arXiv preprint arXiv:2406.13035*, 2024.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

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

Jiaming Yang, Chenwei Tang, Liangli Zhen, and Jiancheng Lv. Rethinking kv cache eviction via a unified information-theoretic objective, 2026. URL https://arxiv.org/abs/2604.25975.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. Turboquant: Online vector quantization with near-optimal distortion rate. *arXiv preprint arXiv:2504.19874*, 2025.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Yuxin Zhang, Yuxuan Du, Gen Luo, Yunshan Zhong, Zhenyu Zhang, Shiwei Liu, and Rongrong Ji. Cam: Cache merging for memory-efficient llms inference. In *Forty-first international conference on machine learning*, 2024.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

Bingxi Zhao, Lin Geng Foo, Ping Hu, Christian Theobalt, Hossein Rahmani, and Jun Liu. Llm-based agentic reasoning frameworks: A survey from methods to scenarios. *arXiv preprint arXiv:2508.17692*, 2025.

A Derivation and analysis of Jacobian Information Capacity

In this appendix, we provide the rigorous mathematical derivation of the Jacobian Information Capacity presented in Theorem 1 of the main text. We begin by explicitly defining the local linearization framework and then derive the closed-form mutual information objective under the resulting Gaussian approximation. Finally, we analyze the structure of the derived Jacobian matrix to provide deeper insights into how it captures the nonlinear dynamics of softmax attention.

A.1 Proof of Theorem 1

Let $\mathcal{Z}_\mathcal{C} = \{(k_i, v_i)\}_{i\in\mathcal{C}}$ be the retained subset of key-value pairs, where $|\mathcal{C}| = m$. Let $K_\mathcal{C} \in \mathbb{R}^{m\times d_k}$ and $U_\mathcal{C} \in \mathbb{R}^{d_o\times m}$ be the stacked key and output-value matrices, respectively. The retained attention mechanism defines a nonlinear mapping $f_\mathcal{C} : \mathbb{R}^{d_k} \to \mathbb{R}^{d_o}$ from a query q to the output y:

$$f_\mathcal{C}(q) = U_\mathcal{C}\,\mathrm{softmax}\left(\frac{K_\mathcal{C} q}{\sqrt{d_k}}\right).$$

We model the attention process as a noisy nonlinear channel:

$$Y_\mathcal{C} = f_\mathcal{C}(q) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \Sigma_{\text{noise}}).$$

We assume future queries are localized around a center μ_Q with covariance Λ_Q, i.e., $q \sim \mathcal{N}(\mu_Q, \Lambda_Q)$. Our goal is to approximate the mutual information $I(q; Y_\mathcal{C}|\mathcal{Z}_\mathcal{C})$.

To make the analysis tractable, we perform a first-order Taylor expansion of the nonlinear mapping $f_\mathcal{C}(q)$ around the query center μ_Q. Let $\delta q = q - \mu_Q$. The expansion is given by:

$$f_\mathcal{C}(q) \approx f_\mathcal{C}(\mu_Q) + J_\mathcal{C}\delta q,$$

where $J_\mathcal{C} = \left.\frac{\partial f_\mathcal{C}(q)}{\partial q}\right|_{q=\mu_Q}$ is the Jacobian matrix evaluated at μ_Q.

We now derive the explicit form of $J_\mathcal{C}$. Let $s(q) = \frac{1}{\sqrt{d_k}} K_\mathcal{C} q \in \mathbb{R}^m$ be the logits, and $\alpha(s) = \mathrm{softmax}(s) \in \mathbb{R}^m$ be the attention weights. By the chain rule:

$$\frac{\partial f_\mathcal{C}}{\partial q} = U_\mathcal{C}\frac{\partial \alpha}{\partial s}\frac{\partial s}{\partial q}.$$

The derivative of the logits with respect to q is $\frac{\partial s}{\partial q} = \frac{1}{\sqrt{d_k}} K_\mathcal{C}$. The derivative of the softmax function $\alpha(s)$ with respect to its input logits s is given by the standard result:

$$\frac{\partial \alpha}{\partial s} = \mathrm{Diag}(\alpha) - \alpha\alpha^\top.$$

Evaluating these derivatives at $q = \mu_Q$, we define the attention weights at the center as $\alpha^\star_\mathcal{C} = \mathrm{softmax}(\frac{K_\mathcal{C}\mu_Q}{\sqrt{d_k}})$, and the softmax Jacobian matrix as $S_\mathcal{C} = \mathrm{Diag}(\alpha^\star_\mathcal{C}) - \alpha^\star_\mathcal{C}(\alpha^\star_\mathcal{C})^\top$. Substituting these back yields the Jacobian of the attention map:

$$J_\mathcal{C} = \frac{1}{\sqrt{d_k}} U_\mathcal{C} S_\mathcal{C} K_\mathcal{C}. \tag{11}$$

This completes the proof of Lemma 1 and provides the explicit form of the effective local linear channel.

Under the first-order approximation, the noisy channel becomes a linear Gaussian channel with respect to the perturbation $\delta q \sim \mathcal{N}(0, \Lambda_Q)$:

$$Y_\mathcal{C} \approx \underbrace{f_\mathcal{C}(\mu_Q)}_{\text{constant bias}} + J_\mathcal{C}\delta q + \epsilon. \tag{12}$$

Since δq and ϵ are independent Gaussian random variables, the output $Y_\mathcal{C}$ is also Gaussian. The mutual information is invariant to the constant bias term. We compute the relevant covariance matrices:

$$\begin{aligned}\Sigma_{Y|q} &= \mathrm{Cov}(Y_{\mathcal{C}}|q) = \mathrm{Cov}(\epsilon) = \Sigma_{\text{noise}}.\\ \Sigma_Y &= \mathrm{Cov}(Y_{\mathcal{C}}) = J_{\mathcal{C}}\,\mathrm{Cov}(\delta q)J_{\mathcal{C}}^\top + \mathrm{Cov}(\epsilon) = J_{\mathcal{C}}\Lambda_Q J_{\mathcal{C}}^\top + \Sigma_{\text{noise}}.\end{aligned} \tag{13}$$

For Gaussian distributions, the mutual information is the difference between marginal and conditional differential entropies, which can be calculated via determinants of covariance matrices:

$$\begin{aligned}\mathcal{L}_{\mathcal{C}}^{\text{Jac}} \approx I(q; Y_{\mathcal{C}}|\mathcal{Z}_{\mathcal{C}}) &= H(Y_{\mathcal{C}}) - H(Y_{\mathcal{C}}|q)\\ &= \frac{1}{2}\log\det(\Sigma_Y) - \frac{1}{2}\log\det(\Sigma_{Y|q})\\ &= \frac{1}{2}\log\frac{\det(J_{\mathcal{C}}\Lambda_Q J_{\mathcal{C}}^\top + \Sigma_{\text{noise}})}{\det(\Sigma_{\text{noise}})}.\end{aligned} \tag{14}$$

Applying the matrix determinant identity $\det(A+B) = \det(A)\det(I + A^{-1}B)$, we obtain the final closed-form objective:

$$\mathcal{L}_{\mathcal{C}}^{\text{Jac}} \approx \frac{1}{2}\log\det\left(I + \Sigma_{\text{noise}}^{-1}J_{\mathcal{C}}\Lambda_Q J_{\mathcal{C}}^\top\right).$$

This completes the derivation of Theorem 1.

A.2 Analysis of the Jacobian Information Capacity

The derived Jacobian capacity differs from prior linear information-theoretic objectives primarily through the structure of the Jacobian matrix $J_{\mathcal{C}} \propto U_{\mathcal{C}}S_{\mathcal{C}}K_{\mathcal{C}}$. The core novelty lies in the **Softmax Competition Matrix** $S_{\mathcal{C}} = \mathrm{Diag}(\alpha_{\mathcal{C}}^\star) - \alpha_{\mathcal{C}}^\star(\alpha_{\mathcal{C}}^\star)^\top$, which explicitly captures the nonlinear dynamics of attention. Here we analyze its two key components:

1. Local Sensitivity and Saturation. The diagonal elements of $S_{\mathcal{C}}$ are given by $S_{ii} = \alpha_i^\star(1-\alpha_i^\star)$, where $\alpha_i^\star$ is the attention weight of the i-th token at the query center μ_Q. This term measures the local sensitivity of the softmax function. $S_{\mathcal{C}}$ offers two new and important insights:

- **Peak Sensitivity:** The term is maximized when $\alpha_i^\star \approx 0.5$, indicating that tokens with moderate attention weights are most sensitive to query perturbations and thus provide the highest potential information gain locally.
- **Saturation Regions:** As $\alpha_i^\star \to 0$ (irrelevant) or $\alpha_i^\star \to 1$ (dominant), the term $S_{ii} \to 0$. This captures the saturation effect: if a token is already ignored or fully dominant, small changes in the query will not significantly change its contribution to the output.

This contrasts sharply with linear models or heuristics that assume higher attention weights always imply higher importance. Our theory suggests that saturated tokens, despite high attention, may have low marginal information value.

2. Token Competition. The off-diagonal elements $S_{ij} = -\alpha_i^\star\alpha_j^\star$ (for $i \neq j$) are always negative. This explicitly models the competition introduced by softmax normalization. An increase in the logit of token j necessarily decreases the attention weights of all other tokens i, coupling their contributions.

The linear gaussian surrogate Yang et al. (2026) effectively assumes $S_{\mathcal{C}}$ is proportional to the identity matrix. By explicitly incorporating $S_{\mathcal{C}}$, the Jacobian Information Capacity rewards retained subsets that are not only aligned with future queries ($K_{\mathcal{C}}$ and $\alpha_{\mathcal{C}}^\star$) but also possess high local sensitivity and distinct output directions ($U_{\mathcal{C}}$) under the constraints of softmax competition.

B More details on JACAP Eviction Method

In this appendix, we detail the principled approximations that reduce the full Jacobian Information Capacity to the efficient, one-shot JACAP scoring rule presented in Algorithm 1. We also provide a complexity analysis of the proposed algorithm.

B.1 Derivation of the JACAP Algorithm

The complexity of the objective $\mathcal{L}_{\mathcal{C}}^{\text{Jac}} \propto \log\det(I + \Sigma_{\text{noise}}^{-1} U_{\mathcal{C}} S_{\mathcal{C}} K_{\mathcal{C}} \Lambda_Q K_{\mathcal{C}}^\top S_{\mathcal{C}}^\top U_{\mathcal{C}}^\top / d_k)$ stems from three coupled structured matrices: the softmax competition matrix $S_{\mathcal{C}}$, the query-key response covariance $K_{\mathcal{C}} \Lambda_Q K_{\mathcal{C}}^\top$, and the subset-dependent selection problem itself. We address these via a series of approximations.

Decoupling Softmax Competition via Diagonal Approximation. The matrix $S_{\mathcal{C}} = \text{Diag}(\alpha_{\mathcal{C}}^\star) - \alpha_{\mathcal{C}}^\star (\alpha_{\mathcal{C}}^\star)^\top$ couples all tokens through its off-diagonal terms. To enable efficient token-wise scoring, we approximate $S_{\mathcal{C}}$ by retaining only its diagonal elements, which capture the local sensitivity of each token's weight to its own logit:

$$S_{\mathcal{C}} \approx D_{\mathcal{C}} := \text{Diag}(\rho_i)_{i \in \mathcal{C}}, \quad \text{where } \rho_i = S_{ii} = \alpha_i^\star (1 - \alpha_i^\star).$$

As analyzed in Appendix A.2, the diagonal term ρ_i perfectly encapsulates the "sensitivity vs. saturation" trade-off. While the off-diagonal terms $-\alpha_i^\star \alpha_j^\star$ model negative pairwise competition, neglecting them is a common and effective simplification in variational inference and large-scale approximations. It preserves the most critical nonlinear effect—that saturated tokens have low local influence—while decoupling token interactions in the sensitivity term.

Under this approximation, the Jacobian becomes $J_{\mathcal{C}} \approx \frac{1}{\sqrt{d_k}} U_{\mathcal{C}} D_{\mathcal{C}} K_{\mathcal{C}}$, and the effective channel matrix inside the log-determinant becomes:

$$J_{\mathcal{C}} \Lambda_Q J_{\mathcal{C}}^\top \approx \frac{1}{d_k} U_{\mathcal{C}} D_{\mathcal{C}} (K_{\mathcal{C}} \Lambda_Q K_{\mathcal{C}}^\top) D_{\mathcal{C}} U_{\mathcal{C}}^\top.$$

Decoupling Query Response via Diagonal Approximation. The term $K_{\mathcal{C}} \Lambda_Q K_{\mathcal{C}}^\top$ is an $m \times m$ matrix representing the covariance of key responses to future queries. Its off-diagonal element $k_i^\top \Lambda_Q k_j$ measures the correlation between the responses of token i and token j. To further decouple the tokens, we approximate this matrix by its diagonal:

$$K_{\mathcal{C}} \Lambda_Q K_{\mathcal{C}}^\top \approx \text{Diag}(\kappa_i)_{i \in \mathcal{C}}, \quad \text{where } \kappa_i = k_i^\top \Lambda_Q k_i.$$

The scalar κ_i quantifies the variance of the i-th token's logit under the future query distribution. A larger κ_i indicates that the token's alignment with queries is highly variable, making it potentially more informative than a token with a constant alignment. Ignoring off-diagonal correlations is equivalent to assuming independent key responses under the query prior, a necessary simplification for efficient scoring that is prevalent in similar analyses.

Combining the two diagonal approximations, the inner term becomes a diagonal matrix of scalar weights:

$$D_{\mathcal{C}} (K_{\mathcal{C}} \Lambda_Q K_{\mathcal{C}}^\top) D_{\mathcal{C}} \approx D_{\mathcal{C}} \,\text{Diag}(\kappa_i) D_{\mathcal{C}} = \text{Diag}(\rho_i^2 \kappa_i)_{i \in \mathcal{C}}.$$

Defining the combined importance weight for token i as

$$w_i = \frac{1}{d_k} \rho_i^2 \kappa_i = \frac{1}{d_k} [\alpha_i^\star (1 - \alpha_i^\star)]^2 (k_i^\top \Lambda_Q k_i),$$

the approximated objective becomes:

$$\mathcal{L}_{\mathcal{C}}^{\text{Jac}} \approx \frac{1}{2} \log\det\left(I + \Sigma_{\text{noise}}^{-1} U_{\mathcal{C}} \,\text{Diag}(w_i)_{i \in \mathcal{C}} U_{\mathcal{C}}^\top\right) = \frac{1}{2} \log\det\left(I + \Sigma_{\text{noise}}^{-1} \sum_{i \in \mathcal{C}} w_i u_i u_i^\top\right). \quad (15)$$

Assuming isotropic noise $\Sigma_{\text{noise}} = \sigma^2 I$ and absorbing constants into w_i, this reduces to the standard form of maximizing the log-determinant of a sum of rank-one matrices.

Leverage Score Selection. Similar to CapKV Yang et al. (2026), instead of combinatorially searching for the optimal subset $\mathcal{C}$, we define a global capacity matrix over the entire candidate pool $\mathcal{H}_t$:

$$A_{\text{Jac}} = I + \sum_{j \in \mathcal{H}_t} w_j u_j u_j^\top. \quad (16)$$

We then select tokens greedily based on their marginal contribution to the log-determinant, which is approximated by the statistical leverage score using the matrix determinant lemma ($\log\det(A + wuu^\top) - \log\det(A) = \log(1 + wu^\top A^{-1}u) \approx wu^\top A^{-1}u$ for small w):

$$\text{score}_i = w_i \cdot u_i^\top A_{\text{Jac}}^{-1} u_i. \tag{17}$$

This final form, used in Algorithm 1, elegantly combines the theoretically derived nonlinear importance weight w_i with the structural diversity term $u_i^\top A^{-1} u_i$, providing a principled and efficient eviction criterion.

B.2 Computational Complexity Analysis

We analyze the computational complexity of JACAP for a single eviction step with N candidate tokens and head dimension $d_h = d_k = d_o$. The computation consists of three stages.

First, the weight computation requires evaluating the attention weights $\alpha_i^\star$, which costs $O(Nd_h)$ for the query-key products and softmax normalization. The query response variance

$$\kappa_i = k_i^\top \Lambda_Q k_i$$

requires $O(Nd_h^2)$ when Λ_Q is full, and reduces to $O(Nd_h)$ under a diagonal approximation. Since we adopt a diagonal covariance in practice, the overall cost of computing w_i is $O(Nd_h)$.

Second, constructing the Jacobian capacity matrix

$$A_{\text{Jac}} = \sum_{j=1}^{N} w_j u_j u_j^\top$$

requires $O(Nd_h^2)$, as each rank-one update costs $O(d_h^2)$. The inversion of the resulting $d_h \times d_h$ matrix further incurs $O(d_h^3)$ complexity. The total cost of this stage is therefore $O(Nd_h^2 + d_h^3)$.

Finally, the leverage-score evaluation

$$u_i^\top A_{\text{Jac}}^{-1} u_i$$

requires $O(d_h^2)$ per token, leading to a total complexity of $O(Nd_h^2)$ for all candidates. Combining all stages, the overall complexity per layer and eviction step is

$$O(Nd_h^2 + d_h^3),$$

which is dominated by the capacity matrix construction and inversion.

Compared with heuristic approaches such as SNAPKV, H2O, KEYDIFF, and KNORM, whose complexity is typically $O(Nd_h)$ or $O(N)$ due to independent per-token scoring, JACAP introduces additional quadratic dependence on the head dimension through inter-token matrix interactions. However, this complexity is shared by other capacity-based approaches such as CAPKV, which also rely on leverage-score computations over a capacity matrix.

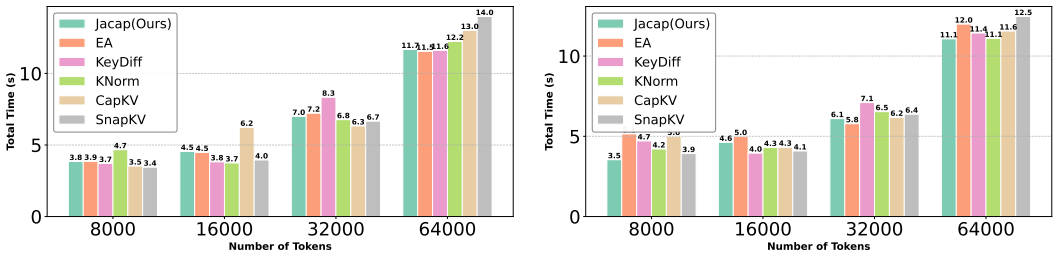
In practice, the additional overhead remains moderate because the attention head dimension is usually small (e.g., $d_h = 128$), and the matrix operations are highly parallelizable on modern GPUs. As demonstrated in our runtime experiments, the improved retention quality in high-compression regimes justifies this additional computational cost.

C More Experiment Results

Our all experiments were conducted on $4\times$ NVIDIA RTX Pro 6000 GPUs with 1024 GB system memory under Ubuntu 22.04.

C.1 Runtime Efficiency Analysis

To evaluate the practical overhead of our proposed method, we confirm the runtime efficiency of JACAP against representative baselines. We measured the total generation time for generating 100 tokens using the Qwen3-8B model on a single NVIDIA RTX pro6000 GPU. The evaluation covers



(a) $C.R. = 0.6$ (b) $C.R. = 0.8$

Figure 4: Runtime comparison of generating 100 tokens with Qwen3-8B on a single RTX 6000 GPU under varying input context lengths and compression ratios (C.R.)

increasing input context lengths from 8k to 64k tokens under two compression ratios (C.R. = 0.6 and 0.8).

The results are summarized in Figure 4. We observe that JACAP exhibits runtime performance that is highly comparable to existing methods across all settings.

The analysis in Appendix B.2 indicates a theoretical complexity of $O(Nd_h^2 + d_h^3)$ for JACAP. These empirical results suggest that in practice, with typical head dimensions (e.g., $d_h = 128$), the matrix operations are efficiently handled by the GPU, making the actual overhead negligible compared to the overall inference process. Therefore, JACAP delivers significant performance gains in high-compression regimes without incurring a prohibitive computational cost.

C.2 MORE EXPERIMENT RESULT ON NIAH

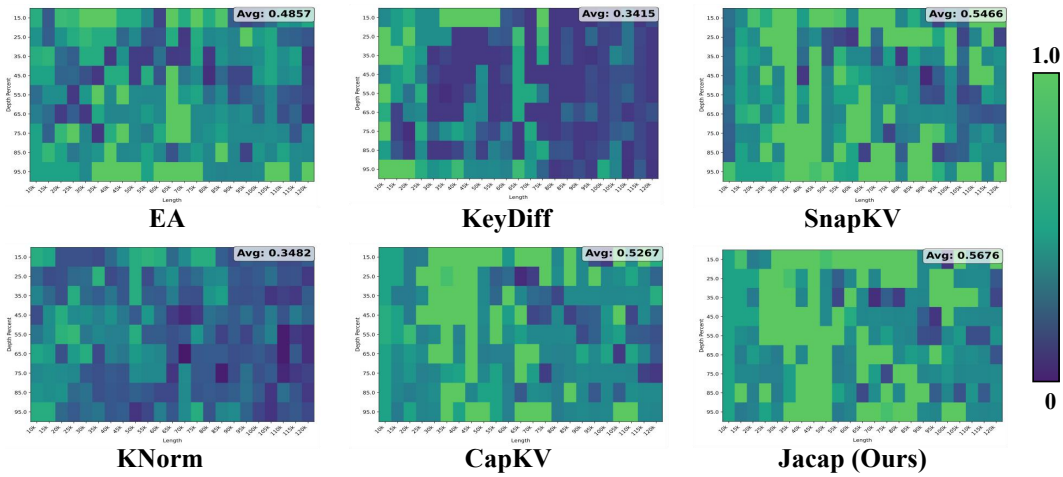


Figure 5: Performance of JACAP and baseline KV-cache eviction methods on the NIAH benchmark at compression ratio 0.5 using Qwen3-8B.

To provide a more comprehensive evaluation across different sparsity regimes, we further evaluate the Needle-in-Haystack (NIAH) performance under a moderate compression ratio of 0.5 using the Qwen3-8B model. The results are shown in Figure 5.

Consistent with the findings under higher compression in the main text, JACAP achieves the best overall performance. The heatmaps reveal that while purely structural methods like KEYDIFF and KNORM fail significantly at long contexts, and attention-based baselines like SNAPKV begin to degrade at extreme lengths (e.g., ¿100k tokens) and deeper positions, JACAP maintains remarkable robustness. It consistently retrieves the needle across the entire spectrum of context lengths and depths, demonstrating that incorporating nonlinear sensitivity effectively identifies and preserves crucial information even when it is sparsely located deep within very long sequences.

C.3 ABLATION OF TEMPERATURE τ

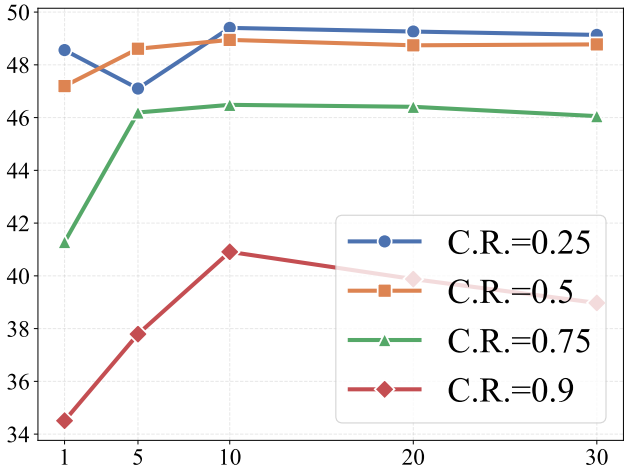


Figure 6: Ablation of temperature τ on LongBench.

Figure 6 reports the average LongBench performance of Qwen3-8B under different temperature values and compression ratios. We observe that JACAP achieves the best overall performance when τ is around 10. A small τ makes the estimated attention prior overly sharp, causing the eviction score to focus excessively on a few high-alignment tokens and potentially discard other informative entries. Conversely, a large τ over-smooths the prior, weakening the query-dependent relevance signal and making the resulting score increasingly dominated by the diversity term. The optimal performance around $\tau = 10$ indicates that a moderate temperature provides a suitable balance between local query relevance and cache diversity.