HYBRIKV: KV CACHE EVICTION FOR HYBRID LINEAR-ATTENTION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent frontier language models increasingly adopt hybrid linear–attention architectures, replacing full attention with linear attention in most layers to reduce long-context memory overhead. However, even when retained in only a few layers, full attention still incurs KV caches that grow linearly with context length, posing a major memory bottleneck for long-context inference. KV-cache eviction provides a complementary way to alleviate this memory pressure, but existing methods are primarily designed for attention-only architectures and do not fully account for the distinct information pathways introduced by hybrid models. We identify two architecture-specific characteristics: the remaining full-attention layers exhibit less concentrated attention, and the linear pathway provides complementary token-level information beyond that available from the full-attention pathway. Motivated by these observations, we introduce HybriKV, a hybrid-aware KV-cache eviction method. To account for the less concentrated attention in the full-attention pathway, HybriKV aggregates evidence from multiple salient responses rather than relying primarily on the strongest one. To exploit complementary token-level information from the linear pathway, it reserves a small portion of the cache budget for positions highlighted by linear-layer signals. Across RULER, LongBench, and SCBench, HybriKV achieves near-lossless performance under approximately 2–3$\times$ KV-cache compression and reduces decoding-time attention latency by 45% compared with FlashAttention. It also consistently outperforms existing KV-eviction methods across hybrid models ranging from 7B to 48B parameters. Code is available in our anonymous repository.

1 INTRODUCTION

Hybrid linear–full attention architectures are increasingly adopted by frontier language models, including recent Qwen (Qwen Team, 2026b), Kimi (Team et al., 2026), and Nemotron (NVIDIA, 2026) model families. By interleaving recurrent-like linear-attention layers with a smaller number of conventional full-attention layers, these models substantially reduce the contextual memory cost of long sequences while retaining global retrieval capability. However, the memory footprint of linear states remains approximately constant with context length, whereas the KV cache in full-attention layers grows linearly, causing the latter to eventually dominate contextual memory at long context, as illustrated for Qwen3.5-9B (Qwen Team, 2026a) in Figure 1(a).

KV-cache eviction is therefore still important for long-context inference with hybrid models. Existing eviction methods, such as SnapKV (Li et al., 2024) and KVzip (Kim et al., 2026b), typically estimate the importance of cached KV entries from attention-derived responses and retain those considered most useful for subsequent computation. This design is natural for attention-only architectures, where contextual information is accessed entirely through the attention pathway. Hybrid architectures, however, introduce a different setting: only a subset of layers retains full attention, and contextual information is additionally propagated through the linear pathway. It is therefore important to understand whether the signals used by existing eviction methods remain equally informative in this new architecture.

We examine these two aspects empirically. First, as shown in Figure 1(b), the remaining full-attention layers in hybrid models exhibit less concentrated attention than their counterparts in attention-only architectures, with higher normalized entropy and lower top-1 mass. This is par-

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.

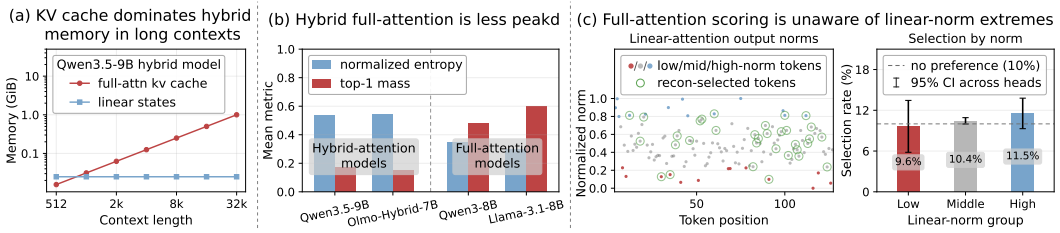

Figure 1: Motivation for hybrid-aware KV-cache eviction. **(a)** Contextual memory breakdown of Qwen3.5-9B. The memory required by linear states is largely independent of sequence length, whereas the KV cache of the remaining attention layers grows with context length and becomes the dominant component at long context. **(b)** Attention concentration differs between hybrid and attention-only architectures. The attention layers in hybrid models show higher normalized entropy and lower top-1 mass, suggesting that salient information is less concentrated in a single dominant response. **(c)** Reconstruction-based eviction does not preferentially preserve tokens highlighted by the linear pathway. We partition tokens into the bottom 10%, middle 80%, and top 10% linear-output norm groups and apply a 10% per-head KV budget. Left: a token-level visualization of one example, where circled positions denote tokens selected by ≥ 5 attention heads. Right: aggregate retention rates over the full evaluation set for the three norm groups. The retention rates of the extreme-norm groups remain close to the budget-induced 10% baseline. Results in (b) and (c) are based on synthetic-text inputs; additional input types are evaluated in Appendix C.1. Panel (c) uses Qwen3.5-9B, with error bars denoting 95% confidence intervals across heads.

ticularly relevant to reconstruction-based KV eviction methods (Kim et al., 2026b; Jegou & Jeblick, 2026; Baek et al., 2026), which have shown strong performance across long-context tasks by estimating cache importance from responses to reconstruction queries and commonly aggregate these responses using their maximum. When attention becomes less concentrated, relying primarily on the strongest response can discard useful evidence distributed across multiple salient responses. Second, the linear pathway exposes additional token-level information that is not reflected by reconstruction responses alone. Figure 1(c) shows that tokens with extreme linear-output norms receive little additional preference under reconstruction-based eviction, with their retention rates remaining close to the budget-induced baseline. Together, these observations indicate that KV eviction for hybrid models should account for both the changed behavior of the remaining full-attention layers and the additional information carried by the linear pathway.

Building on these observations, we introduce **HybriKV**, a hybrid-aware KV-cache eviction framework with two corresponding components. To better capture importance under the less concentrated attention regime, *Mixed Reconstruction Aggregation* combines the strongest reconstruction response with evidence from multiple salient responses, providing a broader view of how each cached position contributes to reconstruction. To incorporate information from the linear pathway, *Linear-Aware Token Protection* identifies a small set of positions indicated by linear-layer signals and protects them with a dedicated portion of the cache budget. Importantly, the two signals retain distinct roles rather than being forced into a shared scoring space. Together, these components adapt reconstruction-based eviction to the two characteristic information pathways of hybrid architectures.

Our main contributions are:

- We revisit KV-cache eviction for hybrid linear–attention models and show that existing attention-centric criteria overlook two architecture-specific properties: less concentrated attention and complementary signals from the linear pathway.
- We propose HybriKV, which addresses these two properties through mixed reconstruction aggregation and linear-aware token protection, combining richer reconstruction evidence with architecture-specific signals while keeping their roles explicitly separated.
- Across RULER, LongBench, and SCBench, HybriKV consistently improves over the evaluated KV-eviction baselines. We further demonstrate its compatibility with learned importance gates, yielding a more efficient variant while preserving most of the accuracy gains.

2 Related Work

KV cache compression and eviction. KV cache compression reduces the memory overhead of long-context inference through techniques such as quantization (Liu et al., 2024; Zhou et al., 2026), merging (Wang et al., 2024; Zweiger et al., 2026), and eviction. Among them, eviction retains only a subset of cached key–value entries according to their estimated importance. Early approaches often derive retention decisions from observed attention statistics. H_2O (Zhang et al., 2023) maintains tokens with high accumulated attention together with recent tokens, while SnapKV (Li et al., 2024) estimates importance from the attention responses of a small observation window near the end of the prompt. More recent work has increasingly emphasized query-agnostic eviction, where cache retention can be determined without conditioning on a specific downstream query. DuoAttention (Xiao et al., 2025) exploits stable head-level functional differences, retaining the full cache only for identified retrieval heads while assigning compact streaming caches to the remaining heads. Expected Attention (Devoto et al., 2025) estimates future importance from the expected attention received by each cached position, whereas KVzip (Kim et al., 2026b) measures the contribution of individual KV entries through context reconstruction. In parallel, Ada-KV (Feng et al., 2026) introduces adaptive head-wise budget allocation, assigning different cache budgets according to the attention characteristics of individual heads. This allocation strategy is complementary to token importance estimation and has subsequently been adopted by recent eviction methods. To reduce the overhead of importance estimation, Fast KVzip (Kim et al., 2026a) and KVzap (Jegou & Jeblick, 2026) further learn lightweight predictors that directly estimate eviction scores during context processing. HybriKV follows this query-agnostic setting, but focuses on an architectural regime largely unexplored by existing eviction methods: hybrid linear–attention models. Rather than directly transferring importance criteria developed for attention-only architectures, we study how KV retention should account for both the changed behavior of the remaining full-attention layers and complementary information exposed by the linear pathway.

Hybrid linear–attention models. Recent language models increasingly interleave linear or recurrent sequence mixers with a smaller number of full-attention layers, as exemplified by recent model families such as Qwen (Qwen Team, 2026a;b), Kimi (Team et al., 2026), and Nemotron (NVIDIA, 2026). Recent work further shows that the placement and interaction of linear and full-attention layers are important to the behavior of hybrid models (Su et al., 2026; Gu et al., 2026), suggesting that the two components should not be viewed as independent modules. While prior work has primarily focused on designing and analyzing hybrid architectures themselves, we study the implications of this interaction for KV-cache eviction, considering both the changed behavior of the remaining full-attention layers and complementary information exposed by the linear pathway.

3 Preliminaries

3.1 KV Cache in Hybrid Linear–Attention Models

Consider a hybrid causal language model with full-attention layers $\mathcal{F}$ and linear-attention layers $\mathcal{L}$. Linear-attention layers summarize historical information into compact recurrent states whose memory is approximately independent of context length. In contrast, each full-attention layer maintains token-wise key–value states, and its KV cache grows linearly with the context length. As shown in Figure 1(a), the remaining full-attention KV cache therefore becomes the dominant source of contextual memory at sufficiently long context. We focus on reducing this component through KV-cache eviction.

3.2 Reconstruction-Based KV Eviction

Reconstruction-based KV eviction has recently shown strong performance across a wide range of long-context tasks, providing an effective way to estimate KV importance without relying on task-specific queries (Kim et al., 2026b; Jegou & Jeblick, 2026; Baek et al., 2026). The key idea is to measure how strongly cached tokens are reused when the context is reconstructed, and retain those that elicit strong reconstruction responses. We therefore adopt reconstruction-based scoring as the foundation of our method.

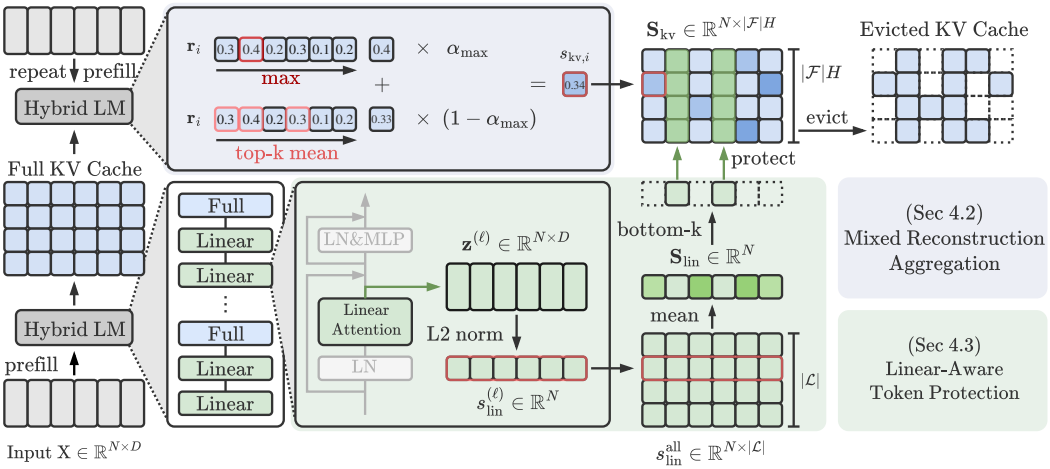

Figure 2: Overview of HybriKV. The original prefill builds the KV cache and collects linear-pathway signals. Context reconstruction produces KV importance scores using Mixed Reconstruction Aggregation. Linear-Aware Token Protection reserves selected token positions, while the remaining cache budget is allocated according to reconstruction scores.

Concretely, the original prefill first constructs the KV cache. A subsequent reconstruction pass performs standard self-attention over the combined original and reconstructed context. For a full-attention layer and one of its KV heads, let $\mathbf{Q}^{\text{rec}}$ denote the queries from the reconstruction pass, and let $\mathbf{K}^{\text{ori}}$ and $\mathbf{K}^{\text{rec}}$ denote the keys from the original and reconstructed context, respectively. The reconstruction attention is computed as

$$\mathbf{A}^{\text{rec}} = \text{softmax}\left(\frac{\mathbf{Q}^{\text{rec}}[\mathbf{K}^{\text{ori}}; \mathbf{K}^{\text{rec}}]^\top}{\sqrt{d}} + \mathbf{M}\right), \tag{1}$$

where $\mathbf{M}$ is the causal attention mask. We extract the responses from reconstruction queries to the original cached keys and define the reconstruction-response vector for position i as

$$\mathbf{A}^{\text{rec}\to\text{ori}} = \mathbf{A}^{\text{rec}}[:, :, \mathcal{I}_{\text{ori}}], \qquad \mathbf{r}_i = \mathbf{A}^{\text{rec}\to\text{ori}}[:, i], \tag{2}$$

where $\mathcal{I}_{\text{ori}}$ indexes the original KV positions. Under grouped-query attention (Ainslie et al., 2023), $\mathbf{r}_i$ collects responses from all query heads associated with the same KV head.

Existing reconstruction-based methods commonly aggregate $\mathbf{r}_i$ using its maximum response,

$$s_{\text{max},i} = \max(\mathbf{r}_i), \tag{3}$$

and rank cached entries according to the resulting importance scores. Conceptually, this max-based criterion follows a conservative retention principle: if any reconstruction query exhibits a strong response to a cached token, that token is considered potentially useful and should be retained.

4 HybriKV

4.1 Overview

HybriKV adapts reconstruction-based KV eviction to hybrid linear–attention models by incorporating information from both the reconstruction and linear pathways. It consists of two complementary components: *Mixed Reconstruction Aggregation* (MRA), which refines the aggregation of reconstruction responses for KV importance estimation, and *Linear-Aware Token Protection* (LATP), which uses signals from the linear pathway to protect a small subset of cached positions.

Figure 2 summarizes the overall pipeline. During the original prefill, we collect linear-pathway signals for token protection. A subsequent reconstruction pass produces importance scores for the cached KV entries, after which the two signals are combined to determine the final retained cache.

4.2 MIXED RECONSTRUCTION AGGREGATION

As described in Section 3.2, reconstruction-based eviction commonly uses the maximum response to determine whether a cached position should be retained. This criterion captures whether any reconstruction query strongly responds to the position, but discards the remaining response distribution. For hybrid models, Figure 1(b) shows that their remaining full-attention layers exhibit less concentrated attention. Motivated by this observation, we complement the maximum response with evidence from multiple strong reconstruction responses.

For cached position i, we compute

$$m_i = \max(\mathbf{r}_i), \qquad t_i = \mathrm{TopKMean}(\mathbf{r}_i), \tag{4}$$

where $\mathrm{TopKMean}(\cdot)$ averages the largest k entries of $\mathbf{r}_i$. We then define the mixed reconstruction score as

$$s_{\mathrm{kv},i} = \alpha_{\max} m_i + (1 - \alpha_{\max}) t_i, \qquad \alpha_{\max} \in [0, 1]. \tag{5}$$

Here, the max term retains sensitivity to the strongest individual response, while the top-k mean captures support from multiple high-response queries. Setting $\alpha_{\max} = 1$ recovers the original max-only scoring rule.

We compute $s_{\mathrm{kv},i}$ independently for each full-attention layer and KV head. Collecting the resulting scores yields $\mathbf{S}_{\mathrm{kv}} \in \mathbb{R}^{N \times |\mathcal{F}| H}$, where H is the number of KV heads per full-attention layer.

4.3 LINEAR-AWARE TOKEN PROTECTION

While reconstruction scores characterize KV importance through the full-attention pathway, hybrid models additionally expose information through their linear layers. As shown in Figure 1(c), such linear-pathway signals are not naturally captured by reconstruction-based selection. We therefore use linear-layer outputs to identify a small set of token positions for explicit protection.

For each linear-attention layer $\ell \in \mathcal{L}$, let $\mathbf{z}^{(\ell)} \in \mathbb{R}^{N \times D}$ denote the output of its attention sublayer during the original prefill. We first compute the per-position output norm and then average it across linear layers:

$$s^{(\ell)}_{\mathrm{lin},i} = \left\| \mathbf{z}^{(\ell)}_{i,:} \right\|_2, \qquad s_{\mathrm{lin},i} = \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} s^{(\ell)}_{\mathrm{lin},i}. \tag{6}$$

Let C denote the number of candidate KV positions and ρ the target eviction ratio. The retained KV budget per head and the portion reserved for linear-aware protection are given by

$$B = (1 - \rho) C, \qquad K_{\mathrm{prot}} = \lfloor \beta B \rfloor. \tag{7}$$

Rather than combining linear and reconstruction scores directly, we use $\mathbf{S}_{\mathrm{lin}} \in \mathbb{R}^N$ only to determine the protected token set. We consider low-norm, high-norm, and two-sided protection, and adopt low-norm protection by default based on the ablation in Section 5.4. Accordingly,

$$\mathcal{P} = \mathrm{BottomK}\left(\mathbf{S}_{\mathrm{lin}}, K_{\mathrm{prot}}\right). \tag{8}$$

The positions in $\mathcal{P}$ are retained across all full-attention layers and KV heads.

4.4 JOINT KV EVICTION

The final cache is constructed in two stages. We first retain the K_{prot} token positions selected by LATP. The remaining capacity is then filled by ranking unprotected KV entries according to the Mixed Reconstruction Aggregation scores $\mathbf{S}_{\mathrm{kv}}$. Following adaptive head-wise budget allocation, different KV heads may retain different numbers of entries according to their reconstruction scores.

The two signals therefore serve complementary roles: the linear pathway identifies a small set of protected token positions, whereas reconstruction scores determine the allocation of the remaining KV capacity. Keeping these mechanisms separate avoids requiring the heterogeneous linear and reconstruction signals to be calibrated into a common scoring space.

Table 1: Main results on RULER ($\rho = 0.7$) and LongBench ($\rho = 0.5$). LongBench results are averaged over six task categories. **Bold** denotes the best average among compressed methods.

	Methods	S1	S2	S3	MK1	MK2	MK3	MV	MQ	VT	CWE	FWE	QA1	QA2	Avg.	S-Doc	M-Doc	Sum.	F-shot	Syn.	Code	Avg.
Qwen3.5-9B	Full KV Cache	100	100	100	99.8	100	100	100	100	100	86.7	99.67	87.6	67.2	95.46	51.40	48.64	22.73	63.9	70.17	41.83	49.76
	SnapKV	98.6	12.6	2.4	15.6	17.6	14.4	10.9	12.35	98.96	82.36	99.33	28.4	40.8	41.10	45.69	43.93	22.25	62.81	69.75	41.15	47.60
	DuoAttention	99.6	99.6	76.8	95.2	14	20.4	50.3	87.1	3.36	21.38	99	31	30.4	56.01	31.66	37.39	20.67	57.04	67.67	37.16	41.93
	Expected Attn.	92.4	88.4	7.6	92	97	11	94.9	96.65	90.32	87.54	90.87	63.2	50.2	74.01	47.18	49.19	22.34	63.21	70.33	34.52	47.79
	KVzip	100	98.8	99.8	98.8	100	96	88.25	99.65	100	93.5	84.2	75.8	60.6	91.95	43.48	50.26	23.12	62.96	61.87	36.52	46.37
	HybriKV	100	98	100	98.8	100	99	98.8	99.7	100	94.58	96	78.4	64.2	**94.42**	45.84	50.95	23.14	62.99	68.37	37.74	**48.17**
Olmo-Hybrid-7B	Full KV Cache	100	100	100	98.6	99.8	97.4	97.25	99.25	95.4	26.64	87.4	72.4	62.4	87.43	41.89	39.56	25.38	45.28	15.55	46.67	35.72
	SnapKV	3.4	71.4	2.4	43.4	6.8	1.2	36.4	41.35	3.44	17.18	77	31	35.2	28.48	38.72	31.74	22.89	43.84	12.09	46.33	32.60
	DuoAttention	100	97	99	58.6	51.8	7	31.8	41.55	31.56	28.22	84.2	29.2	28.6	52.96	21.38	26.61	22.40	43.25	3.28	43.61	26.76
	Expected Attn.	100	99.6	4.4	82.4	80.2	14.8	77.2	87.45	88.48	31.86	86.47	58.6	48.2	66.13	36.47	34.61	23.28	46.35	5.18	48.2	32.35
	KVzip	100	99.6	100	95.8	84.6	17.4	84.15	98.4	83.4	27.22	83.73	68	55	76.72	34.95	36.96	23.73	40.85	11.11	44.39	32.00
	HybriKV	100	100	100	98.4	97.2	90.2	91.85	96.45	82.72	23.02	84.93	76.2	60.8	**84.75**	40.66	39.62	25.09	45.00	11.36	45.13	**34.48**

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Models, benchmarks, and baselines. Our main evaluation uses Qwen3.5-9B (Qwen Team, 2026a) and Olmo-Hybrid-7B (Merrill et al., 2026), covering different full-attention configurations (GQA (Ainslie et al., 2023) vs. MHA (Vaswani et al., 2017)) and positional settings (RoPE-based (Su et al., 2024) vs. RoPE-free (Gelberg et al., 2025)). We evaluate both models on RULER (Hsieh et al., 2024) and LongBench (Bai et al., 2024); RULER uses the 16K setting with 13 tasks, while LongBench contains 21 tasks with an average reported length of approximately 8.8K. We further evaluate Qwen3.5-9B on SCBench (Li et al., 2025), using 11 tasks with an average input length of approximately 112K tokens after excluding Zh.QA, whose approximately 1.5M-token average input length exceeds the context length supported by the model. To assess generality across model scales and hybrid architectures, we additionally evaluate Kimi-Linear-48B-A3B (Team et al., 2025b) and Gemma-3-12B-it (Team et al., 2025a) on both RULER and LongBench. Finally, we evaluate the learned-gate variant of HybriKV on RULER, LongBench, SCBench, and further assess its decoding-time performance on AIME25 (Zhang & Math-AI, 2025) and MATH500 (Hendrycks et al., 2021).

We compare against KVzip (Kim et al., 2026b), Expected Attention (Devoto et al., 2025), SnapKV (Li et al., 2024), and DuoAttention (Xiao et al., 2025), together with the uncompressed reference. Following KVzip, all baselines are evaluated in a query-agnostic setting. Expected Attention is combined with Ada-KV (Feng et al., 2026) for adaptive head-wise budget allocation, and DuoAttention uses KVzip scores for head ranking. For decoding-time experiments, the learned-gate variant of HybriKV is compared with Fast KVzip (Kim et al., 2026a).

Implementation details. Unless otherwise specified, HybriKV uses low-norm protection with $\beta = 0.05$ and sets $\alpha_{\max} = 0.05$. The top-k mean in Mixed Reconstruction Aggregation is computed over the top 5% responses in $\mathbf{r}_i$. More implementation details are provided in Appendix A.1.

5.2 MAIN RESULTS

Table 1 reports the main comparison on RULER and LongBench. Overall, HybriKV consistently achieves the best average performance among the compressed-cache methods on both hybrid models. On RULER at $\rho = 0.7$, HybriKV reaches 94.42 on Qwen3.5-9B, improving over KVzip from 91.95 and approaching the 95.46 full-cache reference. The gain is more pronounced on Olmo-Hybrid-7B, where HybriKV improves the average score from 76.72 with KVzip to 84.75, reducing most of the gap to the 87.43 full-cache result. These improvements are also broadly reflected across the individual retrieval tasks rather than being driven by a single category.

On LongBench at $\rho = 0.5$, where the tasks emphasize broader long-context understanding and question answering, the differences between eviction methods are smaller. Nevertheless, HybriKV achieves the highest compressed-cache average on both models, reaching 48.17 on Qwen3.5-9B and 34.48 on Olmo-Hybrid-7B. The consistent gains across Qwen3.5-9B and Olmo-Hybrid-7B, which differ in both full-attention configuration and positional encoding, suggest that the benefits of hybrid-aware eviction generalize across different hybrid model designs.

Table 2: Longer-context evaluation on 11 SCBench tasks with Qwen3.5-9B.

	Methods	String Retrieval			Semantic Retrieval			Global Information			Multi-Tasking		Avg.
		KV	Prefix-Suffix	MultiHop	RepoQA	EN.QA	MultiChoice	Find	ManyShot	Sum	Sum+NIAH	RepoQA+KV	
	Full KV Cache	99.8	89.4	57.73	7.73	35.33	19.65	50.5	82.59	36.43	68.44	46.45	54.01
$\rho = 0.5$	SnapKV	36.8	25.6	36.49	5.23	30.84	20.1	49.67	81.48	35.39	67.85	3.41	35.71
	DuoAttention	16	41	26.04	4.55	11.11	20.09	42	81.11	32.84	63.34	9.23	31.57
	Expected Attn.	86.4	59	76.49	5.91	33.05	17.47	48.83	85.93	35.83	65.8	38.21	50.27
	KVzip	98.8	87.6	58.84	7.96	33.62	19.21	52.67	74.44	36.78	68.67	41.19	52.71
	HybriKV	98.6	87.4	59.64	7.96	33.05	19.21	53.83	80.37	36.61	68.62	42.47	**53.43**
$\rho = 0.7$	SnapKV	14.8	11	29.47	2.96	27.24	18.14	46.17	86.3	34.09	67.38	2.42	30.91
	DuoAttention	1	1.2	1.73	2.73	3.99	20.52	13.5	87.78	27.37	26.55	1.56	17.08
	Expected Attn.	18.2	3.8	76.27	2.27	30.77	18.34	35.5	85.19	34.09	61.52	7.53	33.95
	KVzip	90.4	73	60.27	7.05	27.35	18.34	40.17	72.22	35.37	62.66	33.67	47.32
	HybriKV	95.8	82.4	59.56	7.5	26.78	19.21	44.67	79.26	36.12	65.69	37.64	**50.42**

Table 3: Learned-gate extension of HybriKV.

	Methods	RULER $\rho = 0.7$	LongBench $\rho = 0.5$	SCBench $\rho = 0.7$	AIME25 4096 token budget	MATH500 1024 token budget	Avg.
Qwen3.5-9B	Full KV Cache	95.46	49.76	54.01	63.33	67.8	66.07
	KVzip	91.95	46.37	47.32	-	-	-
	HybriKV	94.42	48.17	50.42	-	-	-
	Fast KVzip	89.11	46.41	47.12	56.67	64.8	60.82
	Fast HybriKV	92.92	48.08	50.2	63.33	65.8	**64.07**

5.3 Generalization and Extensions

Longer-context evaluation. To examine whether the benefit of hybrid-aware eviction extends to substantially longer contexts, we further evaluate HybriKV on SCBench (Li et al., 2025) using Qwen3.5-9B. Table 2 reports results under 50% and 70% KV cache compression. For the multi-tasking tasks, we report the mean performance across the constituent subtasks of each task. At $\rho = 0.5$, HybriKV achieves the best compressed-cache average of 53.43, improving over KVzip from 52.71 and outperforming the other eviction baselines by a larger margin. When compression becomes more aggressive at $\rho = 0.7$, the advantage becomes substantially clearer: HybriKV reaches 50.42, compared with 47.32 for KVzip and 33.95 or lower for the remaining baselines. This trend indicates that the hybrid-aware retention criterion becomes particularly valuable when the available KV budget is more constrained.

Learned-gate extension. Reconstruction-based importance estimation introduces non-negligible compression overhead because it requires an additional context pass. Existing learned-gate approaches such as Fast KVzip (Kim et al., 2026a) reduce this cost by distilling reconstruction-derived importance scores into lightweight predictors. Following this direction, we instantiate the hybrid-aware retention criterion with learned importance gates, yielding **Fast HybriKV**. As shown in Table 3, Fast HybriKV consistently improves over Fast KVzip across all three long-context benchmarks. The gains are particularly clear on RULER and SCBench, where Fast HybriKV improves the score from 89.11 to 92.92 and from 47.12 to 50.2, respectively. Despite replacing reconstruction-time scoring with learned prediction, Fast HybriKV also preserves most of the performance of reconstruction-based HybriKV.

The learned-gate formulation further enables online KV eviction during decoding. On AIME25 and MATH500, Fast HybriKV outperforms Fast KVzip under fixed decoding-time KV budgets, improving from 56.67 to 63.33 on AIME25 and from 64.8 to 65.8 on MATH500. Overall, Fast HybriKV achieves an average score of 64.07 across the five evaluated settings, compared with 60.82 for Fast KVzip. These results show that the proposed hybrid-aware retention principle extends beyond reconstruction-based context compression and remains effective when importance is predicted dynamically during decoding.

Scaling to a larger hybrid model. To examine whether the gains of HybriKV persist at a larger model scale, we further evaluate Kimi-Linear-48B-A3B (Team et al., 2025b) on RULER and LongBench, using KVzip as a strong reconstruction-based baseline. As shown in Table 4, at moderate

Table 4: Generalization to Kimi-Linear-48B-A3B and Gemma-3-12B-it.

Methods	Kimi-Linear-48B-A3B				Gemma-3-12B-it			
	$\rho = 0.7$		$\rho = 0.85$		$\rho = 0.5$		$\rho = 0.7$	
	RULER	LongBench	RULER	LongBench	RULER	LongBench	RULER	LongBench
Full KV Cache	93.16	50.45	93.16	50.45	86.43	47.68	86.43	47.68
KVzip	92.51	48.77	89.15	42.49	85.22	47.00	82.68	42.53
HybriKV	**92.59**	**49.54**	**91.35**	**45.10**	**85.51**	**47.18**	**83.65**	**44.62**

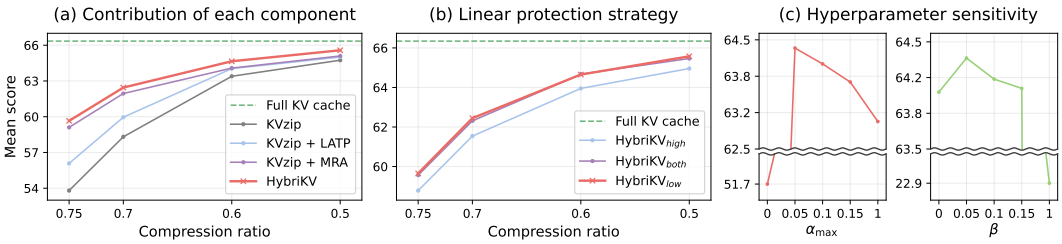

Figure 3: Ablation study on Qwen3.5-9B. Results are averaged across RULER, LongBench, and SCBench. **(a)** Contributions of MRA and LATP. **(b)** Comparison of linear protection strategies. **(c)** Sensitivity to $\alpha_{\max}$ and β. For (c), we use RULER at $\rho = 0.7$, LongBench at $\rho = 0.5$, and SCBench at $\rho = 0.7$; when varying one hyperparameter, the other is fixed to 0.05.

compression ($\rho = 0.7$), both methods preserve most of the full-cache performance, while HybriKV provides consistent improvements on both benchmarks. The gap becomes substantially larger under stronger compression ($\rho = 0.85$), where HybriKV improves over KVzip by 2.20 points on RULER and 2.61 points on LongBench. These results show that the advantage of HybriKV persists at a substantially larger model scale and becomes more pronounced as the KV budget becomes more constrained.

Generalization to a sliding-window/full-attention hybrid model. We further evaluate HybriKV on Gemma-3-12B-it (Team et al., 2025a) to examine whether its benefits transfer beyond linear–full hybrid architectures. Gemma-3-12B-it interleaves sliding-window and full-attention layers, providing a different form of hybrid attention. As shown in Table 4, HybriKV consistently improves over KVzip on both RULER and LongBench, with the improvement becoming larger under stronger compression. At $\rho = 0.7$, HybriKV improves RULER from 82.68 to 83.65 and LongBench from 42.53 to 44.62. These results show that HybriKV can also provide consistent gains on a sliding-window/full-attention hybrid architecture. A detailed component analysis in Appendix B.2 further examines how the individual components of HybriKV behave in this setting.

5.4 Ablation Studies

All ablations are conducted on Qwen3.5-9B, with performance averaged across RULER, LongBench, and SCBench.

Contribution of each component. Figure 3(a) adds Mixed Reconstruction Aggregation (MRA) and Linear-Aware Token Protection (LATP) individually to the KVzip backbone. Both components consistently improve performance across compression ratios. LATP shows larger gains as compression becomes more aggressive, while MRA provides strong improvements throughout the evaluated range. Combining them achieves the best overall performance, supporting their complementary roles in reconstruction-based scoring and linear-aware protection.

Linear protection strategy. Figure 3(b) compares protecting low-norm, high-norm, or both norm extremes. Low-norm and two-sided protection perform similarly and consistently outperform high-norm protection, with low-norm protection giving the best overall average. We therefore use low-norm protection by default. A qualitative analysis in Appendix C.2 further shows that low- and high-norm protection can select qualitatively different token types, providing additional insight into their different behavior.

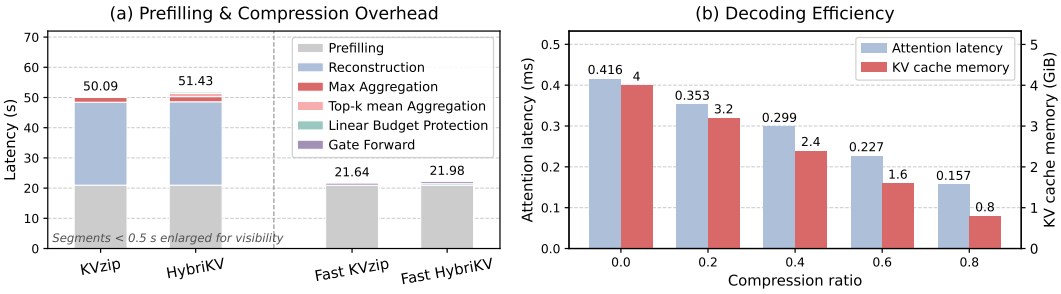

Figure 4: Efficiency analysis on Qwen3.5-9B with a 128K-token context.

Hyperparameter sensitivity. Figure 3(c) varies $\alpha_{\max}$ and β independently. For reconstruction aggregation, a small maximum contribution performs best: $\alpha_{\max} = 0.05$ clearly improves over both top-k-mean-only ($\alpha_{\max} = 0$) and max-only ($\alpha_{\max} = 1$) scoring, indicating that the strongest response is useful when combined with evidence from multiple strong responses. For linear protection, introducing a small protection budget improves performance, with the best average obtained at $\beta = 0.05$. Increasing β beyond this point gradually reduces performance, while $\beta = 1$ causes a sharp degradation. This confirms that linear-aware protection is most effective as a complement to reconstruction rather than as the sole retention criterion. We use $\alpha_{\max} = \beta = 0.05$ by default.

5.5 Efficiency Analysis

We evaluate both context-preparation overhead and decoding-time efficiency on Qwen3.5-9B with a 128K-token context using a single NVIDIA RTX 5090 GPU. For non-uniform head-wise compression, we use variable-length FlashAttention-2 (Feng et al., 2026).

Prefilling and compression overhead. Figure 4(a) compares the context-preparation cost of KVzip, HybriKV, Fast KVzip, and Fast HybriKV. Compared with KVzip, HybriKV introduces only a small additional overhead despite computing the top-k mean and performing linear-aware token protection: the total latency increases from 50.09 s to 51.43 s. The extra cost is minor relative to the reconstruction pass that already dominates context preparation. When combined with learned importance gates, reconstruction-side scoring can be absorbed into gate prediction, substantially reducing the overall preparation cost. Fast HybriKV requires 21.98 s, close to Fast KVzip at 21.64 s while retaining the hybrid-aware retention criterion.

Decoding efficiency. Figure 4(b) reports the effect of KV cache compression on decoding-time attention latency and cache memory. Both quantities decrease steadily as the compression ratio increases. The reported attention latency is the average across all full-attention layers. At $\rho = 0.8$, the KV cache is reduced from 4.0 GiB to 0.8 GiB, while the average attention latency decreases from 0.416 ms to 0.157 ms.

6 Conclusion

Hybrid linear–attention models reduce contextual memory substantially, yet the KV caches of their remaining full-attention layers remain a key bottleneck for long-context inference. We introduce HybriKV, a hybrid-aware KV cache eviction method that accounts for both the changed attention behavior of hybrid models and complementary signals from the linear pathway through Mixed Reconstruction Aggregation and Linear-Aware Token Protection. Across RULER, LongBench, and SCBench, HybriKV preserves near-full-cache performance under substantial KV compression while consistently improving over existing eviction baselines. Its benefits further extend to the larger Kimi-Linear-48B-A3B model, and results on Gemma-3-12B-it indicate that the reconstruction-side design can generalize to other hybrid-attention architectures. We also show that the hybrid-aware design is compatible with learned importance gates, yielding Fast HybriKV with substantially lower compression overhead while retaining most of the accuracy gains. Overall, our results highlight the importance of adapting KV retention to the architecture and information pathways of hybrid models.

AI USE STATEMENT

In this work, we used generative AI tools for writing refinement and assistance with the implementation of selected code components. Generative AI tools were not used for other research tasks requiring disclosure. All AI-assisted text was reviewed and revised by the authors, and all AI-assisted code was inspected, tested, and verified for correctness before use. We take full responsibility for the final content of this work, including all text, claims, experimental results, and code produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed experimental settings, implementation configurations, and additional results in the main paper and appendix to support reproducibility. The complete codebase for this work will be publicly released. For the anonymous submission, the implementation and scripts for reproducing the main experiments are available in our anonymous repository.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 4895–4901, 2023.

Changwoo Baek, Seungjun Shin, and Kyeongbo Kong. Restorekv: Recovering full-cache behavior under aggressive query-agnostic kv cache eviction. *arXiv preprint arXiv:2608.01247*, 2026.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 3119–3137, 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.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S Kevin Zhou. Ada-kv: Optimizing kv cache eviction by adaptive budget allocation for efficient llm inference. *Advances in Neural Information Processing Systems*, 38:113152–113188, 2026.

Yoav Gelberg, Koshi Eguchi, Takuya Akiba, and Edoardo Cetin. Extending the context of pretrained llms by dropping their positional embeddings. *arXiv preprint arXiv:2512.12167*, 2025.

Yuxian Gu, Qinghao Hu, Haocheng Xi, Junyu Chen, Shang Yang, Song Han, and Han Cai. Jet-nemotron: Efficient language model with post neural architecture search. *Advances in Neural Information Processing Systems*, 38:47191–47218, 2026.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

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? *arXiv preprint arXiv:2404.06654*, 2024.

Simon Jegou and Maximilian Jeblick. Kvzap: Fast, adaptive, and faithful kv cache pruning. *arXiv preprint arXiv:2601.07891*, 2026.

Jang-Hyun Kim, Dongyoon Han, and Sangdoo Yun. Fast kvzip: Efficient and accurate llm inference with gated kv eviction. *arXiv preprint arXiv:2601.17668*, 2026a.

Jang-Hyun Kim, Jinuk Kim, Sangwoo Kwon, Jae W Lee, Sangdoo Yun, and Hyun Oh Song. Kvzip: Query-agnostic kv cache compression with context reconstruction. *Advances in Neural Information Processing Systems*, 38:167563–167591, 2026b.

Yucheng Li, Huiqiang Jiang, Qianhui Wu, Xufang Luo, Surin Ahn, Chengruidong Zhang, Amir Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, et al. Scbench: A kv cache-centric analysis of long-context methods. In *International Conference on Learning Representations*, volume 2025, pp. 66063–66093, 2025.

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, 2024.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*, 2024.

William Merrill, Yanhong Li, Tyler Romero, Anej Svete, Caia Costello, Pradeep Dasigi, Dirk Groeneveld, David Heineman, Bailey Kuehl, Nathan Lambert, et al. Olmo hybrid: From theory to practice and back. *arXiv preprint arXiv:2604.03444*, 2026.

NVIDIA. Nemotron 3 ultra: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning, 2026. URL `https://research.nvidia.com/labs/nemotron/files/NVIDIA-Nemotron-3-Ultra-Technical-Report.pdf`. White Paper.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL `https://qwen.ai/blog?id=qwen3.5`.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL `https://qwen.ai/blog?id=qwen3.8`.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Zunhai Su, Bohan Sun, Xialie Zhuang, Shuibai Zhang, He Xiao, Jing Xiong, Hengyuan Zhang, Zhongzhu Zhou, Tiantian Zhang, Ngai Wong, et al. Massive activations in hybrid linear attention large language models: Pre-attention spikes and inter-spike plateaus. *arXiv preprint arXiv:2608.12149*, 2026.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025a.

Kimi Team, Yu Zhang, Zongyu Lin, Xingcheng Yao, Jiaxi Hu, Fanqing Meng, Chengyin Liu, Xin Men, Songlin Yang, Zhiyuan Li, et al. Kimi linear: An expressive, efficient attention architecture. *arXiv preprint arXiv:2510.26692*, 2025b.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Zheng Wang, Boxiao Jin, Zhongzhi Yu, and Minjia Zhang. Model tells you where to merge: Adaptive kv cache merging for llms on long-context tasks. *arXiv preprint arXiv:2407.08454*, 2024.

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*, volume 2025, pp. 37228–37253, 2025.

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

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.

Zhongzhu Zhou, Donglin Zhuang, Jisen Li, Ziyan Chen, Shuaiwen Leon Song, Ben Athiwaratkun, and Xiaoxia Wu. Oscar: Offline spectral covariance-aware rotation for 2-bit kv cache quantization. *arXiv preprint arXiv:2605.17757*, 2026.

Adam Zweiger, Xinghong Fu, Han Guo, and Yoon Kim. Fast kv compaction via attention matching. *arXiv preprint arXiv:2602.16284*, 2026.

A Experiment Details

Table 5: Implementation details of the compared methods.

Method	Core settings	Additional details
SnapKV (Li et al., 2024)	`window_size` $= 64$, `kernel_size` $= 5$	–
DuoAttention (Xiao et al., 2025)	`sink_size` $= 128$, `recent_size` $= 256$	Heads are classified using KVzip (Kim et al., 2026b) scores; the settings apply to streaming heads.
Expected Attention (Devoto et al., 2025)	`n_future_positions` $= 512$, `n_sink` $= 4$, `use_covariance=True`, `use_vnorm=True`, $\epsilon = 0$	Combined with Ada-KV (Feng et al., 2026) for adaptive head-wise budget allocation with $\alpha_{\text{safeguard}} = 0.2$, ensuring at least 20% retention per head.
KVzip (Kim et al., 2026b)	`n_sink` $= 4$, `chunk_size` $= 2048$	–
HybriKV (ours)	`n_sink` $= 4$, `chunk_size` $= 2048$, $\alpha_{\max} = 0.05$, $\beta = 0.05$	Top-k mean uses the top 5% responses in $\mathbf{r}_i$; low-norm protection is used by default.
Fast KVzip (Kim et al., 2026a)	`n_sink` $= 4$, `window_size` $= 4096$, `window_ratio` $= 0.02$, `gate_sink` $= 16$, `gate_dim` $= 16$	The gate is trained on a subset of FineWeb-Edu (Penedo et al., 2024). Compression is performed after prefill and every 128 decoding steps.
Fast HybriKV (ours)	Same gate architecture and compression settings as Fast KVzip	The gate is trained using MRA reconstruction scores as supervision. Linear-aware token protection is applied separately with $\beta = 0.05$.

A.1 Implementation Details

Table 5 summarizes the implementation settings for all compared methods. Unless otherwise stated, we follow the default settings of the corresponding methods and use matched KV cache budgets for fair comparison. For Expected Attention, we combine its importance scores with Ada-KV for adaptive head-wise budget allocation. Fast HybriKV follows the same gate architecture and compression schedule as Fast KVzip; the gate is trained from the mixed reconstruction scores, while linear-aware token protection is applied separately at inference time. Algorithm 1 summarizes the complete KV-eviction procedure of HybriKV.

B Additional Experimental Results

B.1 Task-Level LongBench Results

Table 6 provides the complete task-level LongBench (Bai et al., 2024) results corresponding to the category-level averages reported in the main text. We report all 21 tasks for both Qwen3.5-9B (Qwen Team, 2026a) and Olmo-Hybrid-7B (Merrill et al., 2026), covering multi-document QA, single-document QA, summarization, few-shot learning, synthetic tasks, and code.

B.2 Generalization to Sliding-Window Hybrid Models

Table 7 provides a finer-grained analysis of HybriKV on the sliding-window/full-attention architecture of Gemma-3-12B-it (Team et al., 2025a). The results show that most of the gain over KVzip comes from Mixed Reconstruction Aggregation (MRA). At both compression ratios, KVzip+MRA consistently improves over KVzip on RULER and LongBench.

In contrast, Linear-Aware Token Protection (LATP) alone provides limited benefit in this setting. Here, LATP is instantiated using the output norms of the sliding-window layers. Since each sliding-window token only accesses a local neighborhood, these outputs do not expose the same long-range

Algorithm 1 HybriKV: Hybrid-Aware KV Cache Eviction

Require: Hybrid model f_{LM} with full-attention layers $\mathcal{F}$ and linear-attention layers $\mathcal{L}$; context X; target eviction ratio ρ; MRA weight $\alpha_{\max}$; protection ratio β

Ensure: Retained KV cache $\widehat{\mathrm{KV}}$

$\mathrm{KV} \leftarrow \mathrm{Prefill}(f_{\mathrm{LM}}, \mathrm{X})$ ▷ Original prefill
Initialize $\mathbf{S}_{\mathrm{lin}} \in \mathbb{R}^{N}$
for each linear-attention layer $\ell \in \mathcal{L}$ **do**
  Obtain attention-sublayer output $\mathbf{z}^{(\ell)} \in \mathbb{R}^{N \times D}$
  $s_{\mathrm{lin},i}^{(\ell)} \leftarrow \left\| \mathbf{z}_{i,:}^{(\ell)} \right\|_2, \quad \forall i$
end for
$s_{\mathrm{lin},i} \leftarrow \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} s_{\mathrm{lin},i}^{(\ell)}, \quad \forall i$ ▷ Linear-pathway score
Initialize $\mathbf{S}_{\mathrm{kv}} \in \mathbb{R}^{N \times |\mathcal{F}| H}$
Perform reconstruction pass over X with the original KV cache
for each full-attention layer $l \in \mathcal{F}$ **do**
  for each KV head $h = 1, \ldots, H$ **do**
    for each cached position $i = 1, \ldots, N$ **do**
      Obtain reconstruction-response vector $\mathbf{r}_i^{(l,h)}$
      $m_i^{(l,h)} \leftarrow \max\left(\mathbf{r}_i^{(l,h)}\right)$
      $t_i^{(l,h)} \leftarrow \mathrm{TopKMean}\left(\mathbf{r}_i^{(l,h)}\right)$
      $s_{\mathrm{kv},i}^{(l,h)} \leftarrow \alpha_{\max} m_i^{(l,h)} + (1 - \alpha_{\max}) t_i^{(l,h)}$
    end for
  end for
end for
$B \leftarrow (1-\rho)C, \qquad K_{\mathrm{prot}} \leftarrow \lfloor \beta B \rfloor$
$\mathcal{P} \leftarrow \mathrm{BottomK}\left(\mathbf{S}_{\mathrm{lin}}, K_{\mathrm{prot}}\right)$ ▷ Linear-aware token protection
Retain positions in $\mathcal{P}$ for every $l \in \mathcal{F}$ and KV head h
$\mathcal{U} \leftarrow \{(i,l,h) \mid i \notin \mathcal{P},\, l \in \mathcal{F},\, h = 1, \ldots, H\}$
$K_{\mathrm{rem}} \leftarrow |\mathcal{F}| H \left(B - K_{\mathrm{prot}}\right)$
$\mathcal{R} \leftarrow \mathrm{TopK}\left(\left\{ s_{\mathrm{kv},i}^{(l,h)} : (i,l,h) \in \mathcal{U} \right\}, K_{\mathrm{rem}}\right)$ ▷ Adaptive head-wise allocation
$\widehat{\mathrm{KV}} \leftarrow \mathrm{KV}[\mathcal{P} \cup \mathcal{R}]$
return $\widehat{\mathrm{KV}}$

Table 6: Task-level LongBench results at compression ratio $\rho = 0.5$. The 21 tasks correspond to the six task groups reported in the main text.

	Methods	Multi-doc QA				Single-doc QA				Summarization				Few shot				Synthetic			Code		Avg.
		HPQA	2Wiki	Mus	DuR	MF-En	MF-Zh	Nar	Qasper	Gov	QMSum	MN	VCSum	Trivia	SamSum	TREC	LSHT	PC	PR-En	PR-Zh	LCC	Repo	
Qwen3.5-9B	Full KV Cache	60.68	62.70	46.06	25.10	56.49	67.42	33.97	47.73	32.51	21.58	23.81	13.00	90.31	38.54	78.0	48.75	10.5	100	100	40.52	43.13	49.56
	SnapKV	58.94	56.62	44.14	23.07	47.01	49.96	33.90	44.85	31.99	21.56	22.63	12.81	90.89	38.34	73.0	49.00	11.0	99.75	98.5	40.78	41.52	47.16
	DuoAttention	46.46	31.01	22.40	26.78	44.40	47.07	24.17	33.91	29.16	18.94	21.75	12.82	90.17	35.47	68.75	33.75	9.0	99.0	95.0	33.73	40.58	41.16
	Expected Attn.	59.62	55.91	47.76	25.41	55.78	62.65	32.70	45.62	31.82	21.61	23.37	12.54	90.37	37.71	76.5	48.25	11.0	100	100	28.94	40.09	47.98
	KVzip	58.14	52.39	36.82	26.57	56.42	65.75	32.75	46.09	32.76	22.23	24.03	13.44	89.81	38.29	75.0	48.75	20.0	87.0	78.6	33.95	39.08	46.57
	HybriKV	59.30	55.82	41.94	26.30	57.59	65.90	33.71	46.60	32.80	22.65	23.98	13.14	90.31	38.14	76.5	47.00	19.5	97.12	88.5	36.08	39.40	**48.20**
Olmo-Hybrid-7B	Full KV Cache	58.50	46.33	33.37	29.35	55.04	35.57	30.35	37.27	33.63	25.90	26.35	15.64	84.43	38.18	48.5	10.00	3.0	13.0	30.64	48.34	44.99	35.64
	SnapKV	56.17	41.39	31.74	25.57	44.82	29.45	25.73	26.95	30.08	23.11	23.31	15.03	82.38	38.63	42.5	11.86	2.0	14.5	19.76	49.11	43.54	32.27
	DuoAttention	29.82	28.10	12.20	15.41	38.83	21.94	14.47	31.21	30.53	18.96	26.90	13.20	71.26	29.56	66.0	6.19	1.33	2.5	6.00	45.37	41.85	26.27
	Expected Attn.	50.73	42.65	30.06	22.45	51.37	27.29	27.25	32.52	29.00	23.86	24.76	15.48	85.22	35.80	46.0	18.39	0.0	7.5	8.04	48.55	47.85	32.13
	KVzip	51.31	38.91	28.53	21.05	51.08	38.32	23.56	34.88	30.51	24.51	25.33	14.58	79.68	36.23	44.5	3.00	2.5	5.5	25.33	45.75	43.03	31.81
	HybriKV	55.10	43.75	35.44	28.34	53.23	40.70	26.16	38.38	33.06	25.71	26.60	14.99	79.73	39.94	49.5	10.82	3.0	6.5	24.58	47.10	43.16	**34.56**

complementary information as the linear pathway in linear–full hybrid models. Accordingly, the differences among low-, high-, and two-sided protection are small, and MRA remains the dominant contributor to the overall improvement. These results suggest that the reconstruction-side modification transfers broadly across hybrid-attention architectures, whereas the effectiveness of LATP depends more strongly on the information exposed by the non-full-attention pathway.

Table 7: Component analysis on Gemma-3-12B-it.

	Methods	RULER	LongBench
	Full KV Cache	86.43	47.68
$\rho = 0.5$	KVzip	85.22	47.00
	HybriKV$_{high}$	85.47	47.31
	HybriKV$_{low}$	85.51	47.18
	HybriKV$_{both}$	85.59	47.26
	KVzip + MRA	85.48	47.48
	KVzip + LATP	85.16	46.86
$\rho = 0.7$	KVzip	82.68	42.53
	HybriKV$_{high}$	83.65	44.45
	HybriKV$_{low}$	83.65	44.62
	HybriKV$_{both}$	83.66	44.72
	KVzip + MRA	83.82	44.79
	KVzip + LATP	82.47	42.85

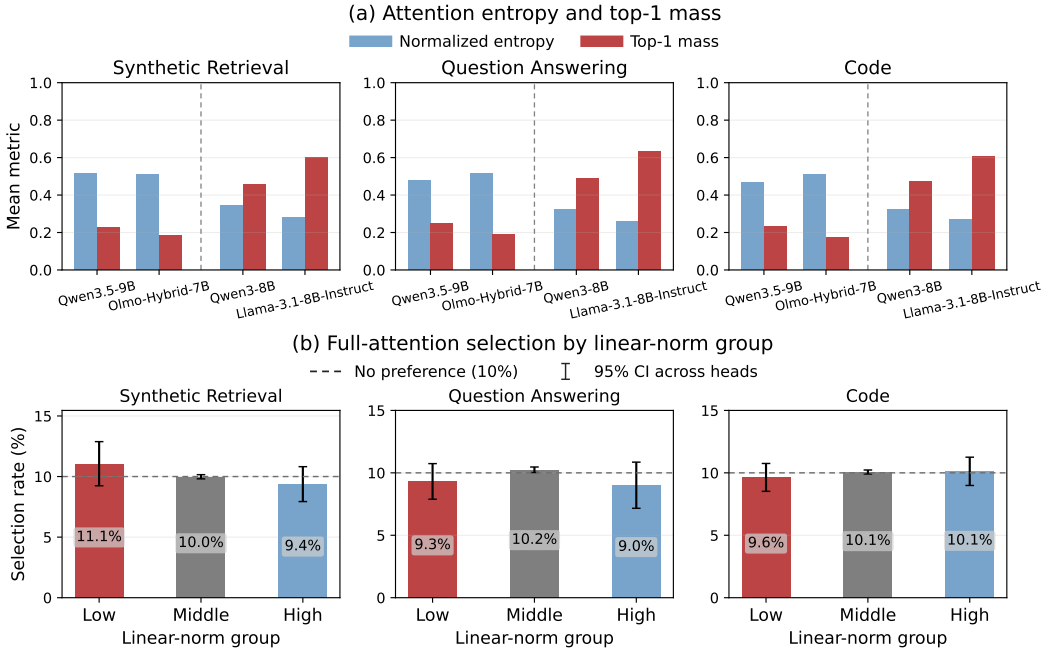

Figure 5: Motivation analysis across different input types. We evaluate synthetic retrieval (`niah_single_3` from RULER), question answering (`multifieldqa_zh` from LongBench), and code (`repobench-p` from LongBench). **(a)** Attention concentration across hybrid and attention-only models, measured by normalized entropy and top-1 mass. **(b)** Reconstruction-based selection rates of the bottom 10%, middle 80%, and top 10% linear-output norm groups on Qwen3.5-9B under a 10% per-head KV budget. The dashed line indicates the budget-induced 10% reference, and error bars denote 95% confidence intervals across heads.

C Additional Analysis

C.1 Robustness of the Motivation Across Input Types

We further examine whether the observations in Figure 1(b,c) persist across different types of inputs. Specifically, we consider three representative settings: *Synthetic Retrieval* using `niah_single_3` from RULER (Hsieh et al., 2024), *Question Answering* using `multifieldqa_zh` from LongBench, and *Code* using `repobench-p` from LongBench.

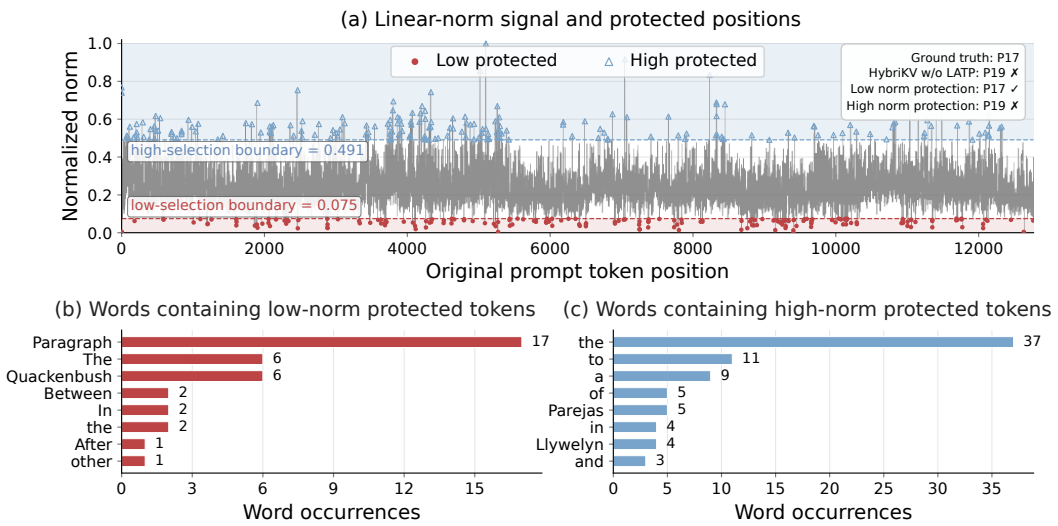

Figure 6: Case study of low- and high-norm token protection on Qwen3.5-9B. **(a)** Normalized linear-output norm over the original prompt and positions selected by low-norm and high-norm protection under the same budget (192 positions each). The ground-truth passage is P17; low-norm protection retrieves P17, while reconstruction without linear protection and high-norm protection retrieve P19. **(b–c)** Most frequent words containing low-norm and high-norm protected tokens, respectively. Tokenizer pieces are reconstructed into words, and each count denotes a word occurrence containing at least one protected token across the 30 passages.

As shown in Figure 5(a), the difference in attention concentration remains consistent across all three input types. The full-attention layers in the hybrid models exhibit higher normalized entropy and lower top-1 mass than those in the attention-only models, indicating that the less concentrated attention behavior observed in Figure 1(b) is not specific to a particular task type.

Figure 5(b) further evaluates reconstruction-based selection with respect to the linear-output norm groups. Under a 10% per-head selection budget, the retention rates of the low-, middle-, and high-norm groups remain close to the 10% reference across synthetic retrieval, question answering, and code inputs. In particular, the corresponding low/middle/high retention rates are 11.1%/10.0%/9.4%, 9.3%/10.2%/9.0%, and 9.6%/10.1%/10.1%, respectively. These results reinforce the observation in Figure 1(c) that reconstruction-based selection provides little systematic preference for the norm extremes exposed by the linear pathway.

C.2 Case Study of Linear-Norm Token Protection

To better understand what information is captured by the linear-pathway signal, we present a qualitative case study from the `passage_retrieval_en` task in LongBench. The input contains 30 passages, and we compare token positions selected by low-norm and high-norm protection under the same protection budget.

Figure 6(a) shows the normalized linear-output norm and the protected positions. Although both strategies use the same protection budget, they select different parts of the sequence. In this example, the ground-truth passage is P17: low-norm protection retrieves P17, whereas both reconstruction without linear protection and high-norm protection retrieve P19.

Figures 6(b) and 6(c) further summarize the words containing protected tokens. Low-norm protection frequently selects passage-structure markers such as “Paragraph” as well as several content-bearing words, whereas high-norm protection is dominated by frequent function words such as “the”, “to”, and “a”, together with a smaller number of named entities. This case illustrates that low- and high-norm positions can encode qualitatively different information, and provides an intuitive example of why low-norm protection can complement reconstruction-based selection. We emphasize that this analysis is qualitative; the aggregate comparison of protection strategies is reported in Section 5.4.