AnchorKV: Training-Free KV Cache Compression via Joint Coverage Diversity and Attention-Based Importance Scoring

Mengbo Wang[1, 2*], **Zhuochen Fan**[2], **Dayu Wang**[3], **Zirui Liu**[4]
Peiyuan Zong[2], **Guorui Xie**[2], **Zeyu Luan**[2], **Qing Li**[2†], **Yong Jiang**[1, 2]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University, China
[2]Pengcheng Laboratory, China [3]Nanyang Technological University, Singapore [4]Peking University, China
wmb25@mails.tsinghua.edu.cn, fanzc@pku.edu.cn, dayu001@e.ntu.edu.sg,
zirui.liu@pku.edu.cn, {zongpy, xiegr, luanzy, liq}@pcl.ac.cn, jiangy@sz.tsinghua.edu.cn

Abstract

KV cache compression is critical for long-context LLM inference, yet existing training-free methods face a coverage–importance dilemma. Attention-based eviction preserves semantically salient tokens but often concentrates the retained cache around a few high-attention regions, creating context holes that hurt long-range reasoning. Diversity-based selection improves key-space coverage but may retain directionally unusual tokens that contribute little to generation. Because future decoding queries are unknown, prefill-time compression must balance directional coverage with attention-based importance. We propose AnchorKV, a training-free prefill-time compression framework that addresses this challenge through block-wise budgeting, anchor-based cosine dissimilarity, causal-attention importance, and a retention-adaptive hybrid score. Experiments on LongBench across three backbone models show that AnchorKV achieves stronger overall performance than existing training-free baselines, especially at low retention ratios. A query-subsampled variant further reduces the overhead of full-query importance estimation by up to 2.7× with a modest average performance drop. We also provide theoretical analysis of query-subsampled importance estimation and block-wise variance reduction.

1 Introduction

Large language models (LLMs) (Touvron et al., 2023; Grattafiori et al., 2024) excel at long-context tasks such as document summarization, retrieval-augmented question answering, and multi-document reasoning. However, standard self-attention (Vaswani et al., 2017) stores Key and Value (KV) representations for every historical token. As context lengths grow to tens of thousands of tokens, the KV cache becomes a major memory and bandwidth bottleneck for LLM serving (Kwon et al., 2023; Sheng et al., 2023). KV cache compression therefore plays a central role in making long-context inference practical.

Existing training-free KV compression methods usually rely on one of two signals: attention-based importance or key-space diversity. **Attention-based eviction** methods, such as H_2O (Zhang et al., 2023) and SnapKV (Li et al., 2024), use accumulated or observed attention weights to retain tokens that appear important. These methods preserve semantically salient tokens, but attention is often highly concentrated around a few local hot spots. When used as a global retention signal, it can allocate too much budget to narrow regions and leave large parts of the context uncovered. **Diversity-based selection** methods, such as KeyDiff (Park et al., 2025), retain tokens that are dissimilar in the Key space. This improves geometric coverage, but key-space dissimilarity alone is not equivalent to semantic relevance: a token may be directionally unusual while carrying little useful information for downstream generation.

We argue that these two failure modes reveal a common challenge in prefill-time KV compression. During prefill, future decoding queries are unavailable, so the compression algorithm cannot know which historical keys will be queried later. The retained KV set must therefore provide broad **directional coverage**: for a range of possible future query directions, the compressed cache should still contain nearby supporting keys. At the same time, coverage alone is insufficient because it can preserve punctuation, formatting artifacts, boilerplate, or other semantically weak outliers. The central challenge is thus not simply to identify important tokens, but to select a compact KV subset that simultaneously preserves directional coverage for unknown future queries and attention-based evidence of current contextual rel-

*Mengbo Wang, Zhuochen Fan and Dayu Wang are co-first authors and contributed equally to this work.
†Corresponding authors: Qing Li and Yong Jiang.

evance.

In this paper, we propose AnchorKV, a training-free and prefill-only KV cache compression framework designed around this coverage–importance objective. First, it reserves structurally important prefix and recent tokens, which protects attention sinks and local generation context. Second, it partitions the compressible region into temporal blocks and assigns each block a local retention budget, preventing global top-k selection from creating context holes. Third, it estimates query-agnostic directional coverage through anchor-based cosine dissimilarity within each block. Fourth, it filters directionally diverse but semantically weak tokens using causal-attention importance, and combines the two signals with a retention-adaptive hybrid score. Our contribution lies in formulating these components as a unified solution to the coverage–importance dilemma, rather than in block partitioning, diversity scoring, or attention importance in isolation.

Our contributions are:

- We identify a coverage–importance dilemma in training-free KV cache compression: attention-only selection preserves salient tokens but creates context holes, whereas diversity-only selection improves coverage but may retain semantically irrelevant tokens.
- We propose AnchorKV, a prefill-only KV compression framework that decomposes token selection into structural protection, temporal coverage, directional coverage, and attention-based filtering.
- We introduce block-wise budgeting and anchor-based diversity scoring to preserve local directional coverage, and combine them with causal-attention importance through a retention-adaptive hybrid score.
- Experiments on LongBench across three backbone models and four baselines show stronger overall performance, especially at low retention ratios. Ablations further show that block-wise budgeting and anchor-based diversity provide complementary gains on top of attention-importance selection.

2 Related Work

The KV cache has become a major bottleneck for long-context LLM inference (Zhang et al., 2023; Xiao et al., 2024). Existing compression methods differ along three axes: *what* to retain, *where* to allocate budget, and *how* to reduce memory or computation. We review attention-based eviction, diversity-based selection, layer/head-level allocation, and recent structured or system-oriented KV optimization methods, and then position AnchorKV.

2.1 Attention-Based Importance Eviction

Attention-based KV eviction retains tokens that receive high attention scores. H_2O (Zhang et al., 2023) accumulates per-token attention weights and preserves heavy hitters. SnapKV (Li et al., 2024) uses an observation window to select tokens likely to be attended to during generation, while Scissorhands (Liu et al., 2023) exploits persistent attention patterns. Recent methods further refine attention-guided eviction. SAGE-KV (Wang et al., 2025) performs self-attention-guided token and head-level eviction after prefilling, and RocketKV (Behnam et al., 2025) combines coarse-grained permanent eviction with decode-time sparse attention.

These methods are effective when observed attention predicts future decoding behavior. However, attention is often concentrated on a few salient spans. Using it as a global retention signal may over-preserve local hot regions while leaving large portions of the context uncovered. AnchorKV complements attention importance with block-wise temporal coverage and anchor-based diversity, aiming to retain broader support for unknown future queries.

2.2 Diversity-Based Selection

Diversity-based methods preserve context coverage by selecting dissimilar tokens. KeyDiff (Park et al., 2025) retains tokens according to Key-space dissimilarity, motivated by its empirical relationship with attention. AnchorKV also uses key-space geometry, but for a different purpose: rather than using dissimilarity as a standalone retention criterion, it treats anchor-based dissimilarity as a query-agnostic coverage signal for unknown future queries. Because pure diversity can retain unusual but semantically weak tokens, AnchorKV combines this coverage signal with attention-based importance through retention-adaptive scoring.

2.3 Layer-Level and Head-Level Budget Allocation

Another line of work studies where to allocate the retention budget. PyramidKV (Cai et al., 2025) assigns larger KV budgets to shallow layers following a pyramidal profile. AdaKV (Feng et al., 2024) and DynamicKV (Zhou et al., 2025) adapt budgets across heads or tasks. These methods are orthogonal to AnchorKV: they decide how much budget each layer, head, or request receives, while AnchorKV decides which tokens to retain within a given budget. Thus, layer/head-level allocation can be combined with AnchorKV's block-wise token selection.

2.4 Structured, Chunk-Level, and System-Oriented KV Compression

Structured methods optimize cache organization or inference execution. StreamingLLM (Xiao et al., 2024) preserves attention sinks and recent tokens; Chelsea (Hu et al., 2025) merg es similar KV pairs to reduce redundancy; POD (Ma et al., 2024) shares keys across adjacent layers. Quest (Tang et al., 2024) performs query-aware sparse attention during decoding, contrasting with prefill-only selection. Recent methods explore related but different axes. ChunkKV (Liu et al., 2025) compresses KV cache at the chunk level by preserving semantically coherent chunks. FastKV (Jo et al., 2026) accelerates long-context processing through token-selective propagation across layers. SCOPE (Wu et al., 2025) separates prefill-stage and decoding-stage KV optimization for long-output generation. PagedEviction (Chitty-Venkata et al., 2026) aligns block-wise pruning with paged memory layouts.

These methods improve efficiency through chunk retention, layer-wise propagation, decode-time sparsity, or system-level memory management. AnchorKV instead performs query-agnostic token-level selection at prefill time. It partitions the context into temporal blocks and selects tokens within each block by jointly considering directional coverage and attention-based importance.

2.5 Quantization and Orthogonal Compression Paradigms

KV cache quantization reduces per-token storage cost rather than token count. KIVI (Liu et al., 2024) introduces tuning-free asymmetric low-bit quantization, and GEAR (Kang et al., 2024) provides a near-lossless KV compression recipe. DiffKV (Zhang et al., 2025) assigns heterogeneous precision to different KV entries. These methods are complementary to token selection: they reduce the cost of retained entries, while AnchorKV decides which entries to keep.

Positioning. AnchorKV is a training-free, prefill-only KV cache compression method that performs block-wise token-level selection before decoding. It avoids attention-only concentration around hot spots, filters semantically weak diversity outliers, and combines directional coverage with attention-based filtering under a fixed retention ratio.

3 Methodology

AnchorKV performs KV cache compression at prefill time. It selects a subset of KV entries before generation and reuses the compressed cache throughout decoding, so selection must be robust to future queries rather than optimized for observed ones.

Figure 1 illustrates the overall pipeline. AnchorKV is built around two constraints. The first is temporal coverage: the retained cache should not collapse into a few high-score regions. The second is attention-based filtering: directionally diverse but weakly supported tokens should not dominate the budget. The following subsections describe how these constraints are implemented through protected zones, block-wise budgeting, anchor-based diversity, causal-attention importance, and hybrid scoring.

Given an input sequence of length L, let $K, V \in \mathbb{R}^{L\times d}$ denote the Key and Value caches of one layer and one KV head, where d is the head dimension. We write $K = [k_1^\top; \ldots; k_L^\top]$ and $V = [v_1^\top; \ldots; v_L^\top]$, where $k_i, v_i \in \mathbb{R}^d$ are the Key and Value vectors of token i. Let $\rho \in (0, 1]$ denote the retention ratio, i.e., the fraction of KV entries kept after compression. AnchorKV outputs a deterministic retained index set $\mathcal{I} \subseteq \{1, \ldots, L\}$, sorted in the original temporal order. The compressed cache is obtained by row selection: $K' = K[\mathcal{I}]$ and $V' = V[\mathcal{I}]$.

The design follows three observations. First, global selection can delete long contiguous regions and create context holes. Second, future queries are unknown at prefill, so the retained keys should cover diverse directions instead of only matching observed attention peaks. Third, directional coverage alone may retain semantically

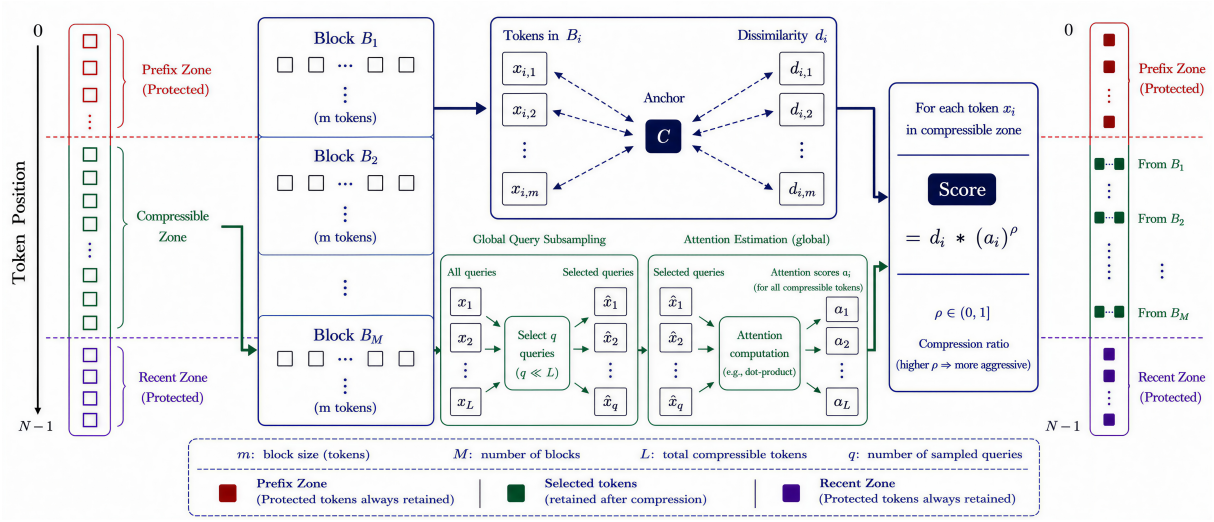

Figure 1: Overview of AnchorKV. The framework is motivated by the coverage–importance dilemma: attention-only selection may over-concentrate retained tokens around a few salient regions, whereas diversity-only selection may preserve directionally unusual but semantically weak outliers. AnchorKV first protects prefix and recent tokens, then partitions the middle context into temporal blocks. Within each block, it computes anchor-based directional diversity and attention-based importance, and selects tokens using a retention-adaptive hybrid score. The selected indices are finally used to form the compressed KV cache.

weak outliers, so an importance signal is still needed.

Sections 3.1–3.5 define protected zones, impose block-wise budgets, estimate directional coverage and attention-based importance, and combine the two scores with a retention-adaptive hybrid rule.

3.1 Context Zoning

Prefix tokens often act as attention sinks (Xiao et al., 2024), while recent tokens directly support local generation. AnchorKV therefore removes these structurally important tokens from ordinary competition for the retention budget. We use fixed token counts rather than ratios because the roles of prefix and recent tokens are structural. Prefix tokens serve as stable global context tokens, while recent tokens preserve local continuity for the first decoding steps. Scaling these zones proportionally with L would unnecessarily consume the retention budget for very long contexts. These protected regions are standard structural safeguards rather than the main proposed contribution. It partitions the KV sequence into three zones:

- **Prefix Zone**: The first L_{prefix} tokens are forcibly retained, where L_{prefix} is a small fixed token count rather than a ratio. In our experiments, we set $L_{\text{prefix}} = 10$.
- **Recent Zone**: The last L_{recent} tokens are also retained, where L_{recent} is a fixed token count. In our experiments, we set $L_{\text{recent}} = 64$.
- **Compressible Zone**: The middle region of length $L_{\text{comp}} = L - L_{\text{prefix}} - L_{\text{recent}}$ contains long-range context and is the primary target of token selection.

3.2 Block Partitioning

Global token selection can delete large context segments, creating *context holes* where errors accumulate along the temporal dimension. Let ϵ_i denote the output perturbation caused by removing or poorly approximating token i in a deleted region $\mathcal{D}$ of the compressible zone. The aggregate perturbation is $E_{\mathcal{D}} = \sum_{i \in \mathcal{D}} \epsilon_i$, and its variance can be decomposed as

$$\mathrm{Var}(E_{\mathcal{D}}) = \sum_{i \in \mathcal{D}} \mathrm{Var}(\epsilon_i) + 2 \sum_{\substack{i<j \\ i,j \in \mathcal{D}}} \mathrm{Cov}(\epsilon_i, \epsilon_j). \tag{1}$$

Here i and j index token positions. When neighboring deleted tokens carry correlated information, large positive covariance terms can amplify the total error. Block-wise budgeting converts global token competition into local competition, ensuring that no short high-score span can consume the entire retention budget.

AnchorKV partitions the compressible zone into temporal blocks $\mathcal{B} = \{B_1, B_2, \ldots, B_M\}$. Each full block contains S tokens, where S denotes the block size; the final block may contain

fewer tokens. After reserving the prefix and recent zones, the total number of retained tokens is

$$n_{\text{kept}} = \lfloor \rho L \rfloor, \tag{2}$$

and the remaining budget for the compressible zone is

$$R_{\text{mid}} = \max\big(n_{\text{kept}} - L_{\text{prefix}} - L_{\text{recent}},\, 0\big). \tag{3}$$

We define the keep ratio inside the compressible region as

$$\rho_{\text{keep}} = \frac{R_{\text{mid}}}{\sum_m |B_m|}. \tag{4}$$

Each block receives a local quota $k_m = \lfloor \rho_{\text{keep}} |B_m| \rfloor$; any residual budget introduced by flooring is assigned to blocks with the largest fractional remainders. This uniform local budgeting preserves temporal coverage and reduces the chance that long-range evidence is removed entirely at low retention ratios. It is a robustness choice rather than a claim of pointwise optimality; the corresponding worst-case interpretation and variance reduction proof are both deferred to Appendix A.

3.3 Anchor-Based Directional Diversity

During decoding, queries may probe the context from many possible directions. Since these future queries are unavailable during prefill, AnchorKV uses a query-agnostic proxy for directional coverage.

Directional coverage diagnostic. Given a retained set of normalized keys $\{\hat{k}_i\}_{i \in \mathcal{S}}$ and T random sign probe vectors $u_t \in \{-1, +1\}^d$, we define a diagnostic coverage score:

$$\text{Cover}(\mathcal{S}) = \min_{t \in \{1,\dots,T\}} \max_{i \in \mathcal{S}} \text{CosSim}(u_t, \hat{k}_i). \tag{5}$$

The score approximates worst-case directional support and is used only for analysis.

Anchor-Based Proxy. Maximizing the above coverage objective exactly is combinatorial. AnchorKV therefore uses a linear-time block-anchor proxy inside each block. For a block B_m, we compute a block anchor μ_m as the mean of normalized keys:

$$\mu_m = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{k}_i. \tag{6}$$

For token $i \in B_m$, its directional diversity score D_i is defined as cosine dissimilarity from the block anchor:

$$D_i = 1 - \text{CosSim}(\hat{k}_i, \mu_m). \tag{7}$$

A high D_i means that the token points away from the local mean direction of its block and can expand the angular span of the retained set. The single anchor is a query-agnostic, linear-time proxy for local directional deviation rather than an exact solution to the set-level coverage objective.

3.4 Attention-Based Importance Estimation

Directional diversity alone can over-select geometric outliers that do not contribute to generation. AnchorKV therefore estimates an attention-based importance score I_i for each token i using a causal attention pass. From hidden representations we obtain query states $Q \in \mathbb{R}^{L \times d}$ and key states $K \in \mathbb{R}^{L \times d}$ for one layer and one KV head, apply RoPE (Su et al., 2024), and compute causal attention:

$$A^{\text{causal}} = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d}} + M\right), \tag{8}$$

where M is the causal mask with $M_{j,i} = -\infty$ for $i > j$. The importance of token i is the average attention it receives from valid future positions:

$$I_i = \frac{1}{u_i} \sum_{j=i}^{L} A^{\text{causal}}_{j,i}, \tag{9}$$

where $u_i = L - i + 1$ is the number of valid causal query positions for key i. For GQA models (Ainslie et al., 2023), we average across query heads within each KV group. Prefill attention is not assumed to predict the requirements of future decoding queries. Instead, it serves as an observable, context-dependent prior that filters geometrically distinctive tokens receiving little support from the current context, while the diversity score preserves directional coverage for unknown future queries.

Query-subsampled approximation. Full causal attention costs $\mathcal{O}(L^2)$. To reduce prefill overhead, AnchorKV uniformly samples query positions $\Omega \subseteq \{1, \dots, L\}$ with $|\Omega| = m = \max(1, \lfloor \rho_q L \rfloor)$, where $\rho_q \in (0, 1]$ is the query sampling ratio. Let $\tilde{A}^{\text{causal}}$ denote the masked causal attention computed on these sampled query rows. The approximate importance is

$$\tilde{I}_i = \text{Avg}_{j \in \Omega,\, j \geq i} \tilde{A}^{\text{causal}}_{j,i}. \tag{10}$$

If no sampled query can observe token i, its sampled importance is set to zero before score normalization. The concentration bound for this query-subsampled estimator is provided in Appendix A.

3.5 Retention-Adaptive Hybrid Scoring

AnchorKV combines directional diversity and attention-based importance with a multiplicative score. For token i, the final score is

$$S_i = D_i \cdot I_i^{\alpha}, \tag{11}$$

where D_i is the anchor-based directional diversity score, I_i is the causal-attention importance score, and α controls how strongly attention-based importance modulates diversity. When $\alpha = 0$, the score reduces to diversity-only selection. Larger α increasingly suppresses tokens with low attention-based importance.

The multiplicative form is intentionally conservative: a token receives a high final score only if it contributes directional complementarity and is not strongly downweighted by attention-based importance. Thus, diversity and importance are not treated as two independent rankings; instead, attention importance acts as a contextual filter on top of query-agnostic coverage. In our default implementation, we set $\alpha = 1-\rho$. This simple retention-adaptive rule strengthens attention-based filtering when the retained budget is small, while allowing the diversity term to play a larger role when more tokens can be kept.

Algorithm 1 summarizes the prefill-time selection procedure.

Overall, AnchorKV converts prefill-time KV compression from global single-score eviction into constrained retention that jointly preserves structural tokens, temporal coverage, directional coverage, and contextual relevance.

The full theoretical analysis, including the estimation error bound, variance reduction, and time complexity, is deferred to Appendix A.

Algorithm 1 AnchorKV: Training-Free KV Cache Compression (single layer).

Require: $(K, V) \in \mathbb{R}^{L\times d}$, retention ratio ρ, block size S, query sampling ratio ρ_q
Ensure: Compressed cache (K', V')

$n_{\text{kept}} \leftarrow \lfloor \rho L \rfloor$
Retain prefix (L_{prefix}) and recent (L_{recent}) zones
$R_{\text{mid}} \leftarrow \max(n_{\text{kept}} - L_{\text{prefix}} - L_{\text{recent}}, 0)$
Compute attention-based importance $\{I_i\}$ using $m = \max(1, \lfloor \rho_q L \rfloor)$ sampled queries
Partition the compressible zone into blocks $\mathcal{B} = \{B_1, \ldots, B_M\}$ of size at most S
$\rho_{\text{keep}} \leftarrow R_{\text{mid}} / \sum_m |B_m|$, $\alpha \leftarrow 1 - \rho$
for each block $B_m \in \mathcal{B}$ **do**
    $\mu_m \leftarrow \frac{1}{|B_m|} \sum_{i\in B_m} \hat{k}_i$
    $D_i \leftarrow 1 - \mathrm{CosSim}(\hat{k}_i, \mu_m)$
    $k_m \leftarrow \lfloor \rho_{\text{keep}} |B_m| \rfloor$ plus residual-budget adjustment if assigned
    $\mathrm{sel}_m \leftarrow \mathrm{TopK}_{i\in B_m}(D_i \cdot I_i^{\alpha},\ k_m)$
end for
$\mathcal{I} \leftarrow \text{prefix} \cup (\bigcup_m \mathrm{sel}_m) \cup \text{recent}$
$(K', V') \leftarrow (K[\mathcal{I}], V[\mathcal{I}])$

4 Experiments

We evaluate AnchorKV on LongBench (Bai et al., 2024) with three goals. First, we test whether joint coverage–importance selection improves over attention-based and diversity-based baselines under different retention ratios. Second, we examine how block-wise budgeting and anchor-based diversity contribute through ablation. Third, we measure whether query-subsampled importance estimation improves prefill efficiency without changing the selection framework.

4.1 Experimental Setup

Benchmark. We evaluate AnchorKV on **LongBench** (Bai et al., 2024), which covers long-context Document QA, Multi-document QA, Summarization, Few-shot learning, Synthetic retrieval, and Code understanding tasks. Additional results on LongBench-v2 (Bai et al., 2025) are reported in Appendix C.

Models. We use public Hugging Face checkpoints for all three backbone models: LLaMA-3.1-8B (Grattafiori et al., 2024), Qwen2.5-7B (Yang et al., 2024), and Qwen3.5-9B (Qwen Team, 2026). All methods are evaluated with the same model weights.

Baselines. We compare against representative training-free KV compression baselines: H_2O (Zhang et al., 2023), StreamingLLM (Xiao et al., 2024), SnapKV (Li et al., 2024), and KeyDiff (Park et al., 2025). An additional comparison with AdaKV-SnapKV (Feng et al., 2024) is reported in Appendix C.

Implementation Details. All methods share the KVPress (Devoto et al., 2025) infrastructure with pure-prefill compression. AnchorKV is evaluated in Full (ρ_q=1.0) and Approx (ρ_q=0.1) variants at retention ratios $\rho \in \{0.1, 0.2, 0.3\}$. Unless otherwise specified, we use $L_{\text{prefix}} = 10$, $L_{\text{recent}} = 64$, and block size $S = 4096$. These values are fixed across all models, tasks, and retention ratios, and are not tuned per benchmark. We will release the AnchorKV implementation, evaluation scripts, configurations, and processed results upon publication.

4.2 Main Results

Table 1 compares KV compression methods on LongBench at retention ratios $\rho \in \{0.1, 0.2, 0.3\}$.

Experimental results support three conclusions.

- **AnchorKV improves over attention-based eviction.** Compared with H_2O and SnapKV, AnchorKV achieves stronger results across most retention ratios and backbone models. This indicates that attention importance alone is insufficient for prefill-time KV selection, because it tends to over-concentrate retained tokens around high-attention regions rather than preserving broad context support.

- **AnchorKV improves over diversity-only selection.** AnchorKV also outperforms KeyDiff in most model–retention settings, showing that key-space dissimilarity should not be used as a standalone retention criterion. Diversity is useful, but it must be paired with attention-based filtering to avoid retaining directionally unusual but unhelpful tokens.

- **The approximate variant preserves most of the gain.** AnchorKV (Approx) generally stays close to AnchorKV (Full), suggesting that query-subsampled importance estimation provides a practical efficiency–quality trade-off while keeping the same coverage–importance selection principle.

4.3 Approximation Analysis

Table 2 compares the prefill latency of the Full (ρ_q=1.0) and Approx (ρ_q=0.1) variants on the LongBench GovReport task (Llama-3.1-8B, ≈8K tokens). The Approx variant reduces prefill latency to 0.37× of Full, corresponding to a 2.7× speedup relative to full-query importance estimation, while Table 1 shows a limited average performance drop. This result indicates that attention-based importance does not require a full-query attention pass to remain useful for AnchorKV's hybrid selection.

We further compare end-to-end wall-clock and inference time on HotpotQA at a 90% compression ratio. Table 3 shows that the Full variant incurs substantial computational overhead from full causal-importance estimation. The Approx variant reduces wall-clock time from 508.8s to 179.9s and is close to AdaKV-SnapKV, but remains slower than SnapKV and StreamingLLM. We therefore treat Full as the accuracy-oriented variant and Approx as the practical accuracy–efficiency trade-off rather than claiming uniformly lower runtime than existing baselines.

4.4 Ablation Study

Table 4 presents the ablation results of AnchorKV's individual modules on Qwen3.5-9B ($\rho = 0.2$):

We also evaluate two compact alternatives to the default scoring rule. On Qwen2.5-7B at 90% compression, replacing multiplicative fusion with a basic additive fusion reduces the average LongBench score from 38.53 to 35.29. At 70% compression, replacing the adaptive exponent ($\alpha = 0.7$) with a fixed $\alpha = 0.9$ reduces the average score from 43.63 to 42.78. These results support both the conservative multiplicative fusion and the retention-adaptive exponent.

The ablation shows that block-wise budgeting and anchor-based diversity act as complementary plug-in constraints on top of the attention-importance backbone. The "w/o Diversity" variant keeps local budgeting but removes anchor-based diversity; its competitive but lower scores indicate that block-wise budgeting alone is useful but does not fully preserve directional complementarity. The "w/o Block Partition" variant keeps hybrid scoring but performs global selection; its larger drops on long-range tasks show that diversity scoring benefits from local budget constraints. Together, these results suggest that each component independently strengthens attention-importance selection, while their combination gives the strongest overall performance.

		Single Doc. QA			Multi Doc. QA			Summarization			Few-shot			Synthetic		Code	
		NarrQA	Qasper	MF-en	HotpotQA	2WikiMQA	Musique	GovRep	QMSum	MultiNews	TREC	TriviaQA	SAMSum	PCount	PR-en	Lcc	RB-P
Llama3.1-8B		30.49	47.36	56.21	59.13	50.82	32.34	35.37	25.04	27.14	28.50	84.31	39.01	10.64	100.00	53.70	47.69
H2O	0.1	27.56	29.68	31.00	45.44	30.60	22.24	28.88	22.00	23.23	26.50	90.20	37.24	8.00	10.50	53.08	48.57
	0.2	32.43	34.96	38.96	51.59	37.77	27.66	31.36	23.04	25.02	7.00	86.18	39.26	9.65	29.50	55.55	49.24
	0.3	32.76	39.88	46.13	55.13	45.62	28.07	32.60	23.91	25.67	10.53	85.49	38.41	9.73	42.50	54.53	50.88
StreamingLLM	0.1	21.40	18.37	22.43	36.36	21.86	14.05	24.95	19.08	19.74	36.25	90.60	34.58	4.00	16.00	52.97	52.59
	0.2	22.81	22.63	24.84	41.52	25.48	18.32	28.03	20.11	22.46	44.00	90.99	35.38	6.00	28.50	53.48	51.12
	0.3	24.34	27.35	26.61	43.74	29.35	21.83	29.76	21.06	24.34	44.50	92.19	36.42	6.00	36.50	52.03	50.45
SnapKV	0.1	26.04	21.04	24.25	45.82	24.38	19.38	25.36	19.90	20.56	34.00	81.08	38.59	6.50	58.00	52.94	48.29
	0.2	28.58	27.02	34.56	51.83	38.97	26.51	28.46	21.85	22.46	34.50	83.66	41.77	10.10	88.00	54.47	48.02
	0.3	28.41	32.56	39.81	55.29	48.12	29.20	30.31	22.80	23.89	36.00	82.71	41.64	9.05	94.50	54.85	46.91
KeyDiff	0.1	25.28	18.18	31.11	42.33	25.09	16.08	26.07	21.49	19.29	33.50	76.01	39.55	9.00	66.00	46.72	47.57
	0.2	30.06	26.81	41.58	49.64	32.40	22.70	28.54	22.70	21.78	37.00	81.61	39.78	9.63	96.00	50.92	46.59
	0.3	28.90	33.93	46.86	52.18	38.96	25.74	30.58	23.00	23.62	38.50	83.94	39.45	11.13	99.00	51.14	46.53
AnchorKV (Full)	0.1	30.28	35.58	36.38	52.04	38.39	28.66	30.63	22.81	23.44	33.00	83.59	40.35	11.03	26.50	55.83	50.83
	0.2	31.35	37.95	46.38	54.49	46.80	29.08	32.65	23.89	25.26	12.50	86.13	34.52	13.25	37.50	56.87	50.67
	0.3	30.53	41.04	49.18	55.43	47.12	28.61	33.38	24.37	25.77	15.00	86.35	34.45	9.38	47.50	55.56	50.14
AnchorKV (Approx)	0.1	30.07	32.04	35.53	51.58	34.47	27.23	29.60	22.91	23.48	28.50	83.66	40.39	12.60	22.00	55.80	49.83
	0.2	31.39	38.47	45.22	55.10	44.39	30.46	31.82	23.79	25.27	10.50	85.33	39.89	13.22	37.00	55.32	49.40
	0.3	31.42	40.58	50.03	54.79	48.35	30.31	32.64	24.41	25.76	18.50	86.41	37.75	12.60	45.50	55.43	49.00
Qwen2.5-7B		29.02	46.13	50.41	55.93	43.26	26.92	33.96	24.20	25.31	68.00	87.83	42.10	13.50	100.00	69.44	66.26
H2O	0.1	24.18	24.12	25.16	36.10	21.76	15.21	29.04	20.41	20.86	48.75	71.32	37.80	6.00	11.50	62.12	58.13
	0.2	25.83	32.42	31.18	44.29	28.65	22.64	31.87	22.30	22.91	57.50	82.28	39.96	6.00	22.50	67.81	63.51
	0.3	25.97	35.93	35.09	50.52	32.58	24.14	32.30	23.20	23.77	63.00	87.71	41.17	5.50	42.50	68.42	64.88
StreamingLLM	0.1	20.89	17.46	22.73	24.86	15.81	7.68	25.55	19.14	16.77	45.50	62.75	34.97	2.00	15.00	55.89	62.18
	0.2	21.87	21.32	25.21	31.09	20.30	8.82	28.23	20.03	20.42	54.17	66.52	36.60	3.00	28.50	58.66	61.72
	0.3	22.95	24.33	27.10	31.39	25.06	11.66	30.34	20.68	22.01	59.00	73.88	37.98	4.50	35.33	60.35	61.61
SnapKV	0.1	19.93	13.88	24.66	38.12	24.18	17.24	26.23	18.37	18.70	41.25	87.55	40.44	6.50	46.50	63.33	62.05
	0.2	21.96	22.16	30.85	46.94	29.10	21.50	29.11	20.23	21.11	50.50	88.58	40.78	6.50	80.50	68.08	62.62
	0.3	24.45	27.67	34.81	50.79	34.77	23.39	30.39	21.06	22.67	56.50	88.00	41.27	8.50	94.17	69.35	63.03
KeyDiff	0.1	19.73	14.81	31.19	32.30	24.55	15.61	25.92	20.03	16.49	40.00	88.64	39.59	3.62	45.67	56.56	62.29
	0.2	23.45	22.74	38.55	45.80	30.67	23.78	28.47	21.56	19.32	55.50	88.92	40.25	7.00	80.67	63.66	62.51
	0.3	23.97	29.74	38.74	49.47	35.22	22.79	29.72	22.02	20.95	64.50	87.77	40.31	7.50	96.33	65.69	63.25
AnchorKV (Full)	0.1	27.43	26.74	29.79	48.37	31.48	24.97	30.50	21.97	20.94	59.00	87.91	40.29	7.00	33.17	63.75	63.14
	0.2	26.49	34.32	37.59	53.92	33.92	28.81	32.11	23.14	23.21	64.00	87.28	40.76	8.50	46.17	68.60	63.92
	0.3	28.16	39.88	41.86	55.85	36.30	30.18	32.79	23.90	23.81	67.00	87.72	40.70	9.00	65.75	69.31	64.36
AnchorKV (Approx)	0.1	27.83	25.42	29.12	42.83	26.81	21.46	30.16	22.07	20.86	55.00	86.39	40.52	6.50	30.17	63.75	62.71
	0.2	28.42	33.12	37.59	51.09	30.19	24.12	31.77	23.28	23.02	61.00	88.25	40.88	8.50	44.25	67.26	63.23
	0.3	26.49	37.87	40.66	54.77	34.89	27.03	32.83	23.72	23.71	64.50	88.27	41.79	10.50	57.33	69.08	64.68

Table 1: LongBench results under different KV retention ratios.

Method	q Ratio	Wall Time (s)	Normalized
Full Queries	1.0	497	1.00
Sampled Queries	0.1	183	0.37
	0.2	224	0.45
	0.3	254	0.51
	0.4	287	0.58

Table 2: Prefill-stage latency comparison between full-query and query-subsampled importance estimation on LongBench GovReport (Llama-3.1-8B, average input length ≈8K tokens). All measurements conducted on a single NVIDIA H200 GPU with batch size 1.

Method	Wall Clock (s)	Inference Time (s)
No Compression	138.751	134.533
KeyDiff	137.853	133.722
SnapKV	160.390	156.246
StreamingLLM	169.299	165.124
AnchorKV (Approx)	179.905	175.708
AdaKV-SnapKV	180.515	176.361
AnchorKV (Full)	508.805	504.706
H_2O	514.561	510.425

Table 3: Wall-clock comparison on HotpotQA at a 90% compression ratio.

4.5 Coverage–Performance Relationship

Block-wise partitioning enforces temporal coverage and ensures that every context segment contributes representative tokens. The ablation study (Table 4) shows that removing block partitioning causes the largest degradation on tasks requiring distant evidence, confirming that this coverage structure is a core part of the method.

To analyze the relationship between directional coverage and downstream performance, we vary $S \in \{128, 256, 512, 1024, 2048, 4096, 8192, 16384\}$ across all 16 LongBench subtasks and three models (Llama-3.1-8B, Qwen2.5-7B, Qwen3.5-9B) at $\rho = 0.2$. Here S is the block size used by the selection procedure. For each (S, mode) configuration, we compute the average directional coverage ratio $\text{Cover}(\mathcal{S}_{\text{kept}})/\text{Cover}(\mathcal{S}_{\text{full}})$ using 2,048 random sign probes, averaged across all layers and samples, as well as the average LongBench score across models. Table 5 reports both metrics jointly.

The analysis reveals three patterns. First, coverage alone is insufficient: the dissim-only mode achieves the highest coverage ratios (0.903–0.905)

Variant	Single Doc. QA			Multi Doc. QA			Summarization			Few-shot			Synthetic		Code	
	NarrQA	Qasper	MF-en	HotpotQA	2WikiMQA	Musique	GovRep	QMSum	MultiNews	TREC	TriviaQA	SAMSum	PCount	PR-en	Lcc	RB-P
Full AnchorKV	22.86	19.57	27.06	40.84	37.17	24.43	24.74	18.13	13.22	52.75	87.88	31.19	7.50	6.50	16.26	57.47
w/o Diversity	20.02	19.09	26.94	39.72	36.92	23.30	23.84	17.91	12.40	51.75	86.71	31.01	7.50	7.50	15.41	57.44
w/o Block Partition	21.89	18.80	24.59	37.67	36.56	21.27	22.82	17.82	11.14	52.25	87.56	31.04	7.50	7.50	13.94	57.30

Table 4: Ablation study of AnchorKV components on Qwen3.5-9B ($\rho = 0.2$). The two key proposed components—diversity weighting and block-wise partitioning—are ablated independently. "w/o Diversity" removes the anchor-based diversity weighting term, reducing to importance-only selection (i.e., causal attention importance with block partitioning). "w/o Block Partition" performs global selection with hybrid diversity–importance scoring.

Mode	Metric	S=128	S=256	S=512	S=1024	S=2048	S=4096	S=8192	S=16384
Both (hybrid)	Cover.	0.899	0.899	0.899	0.899	0.899	0.898	0.896	0.889
	Score	48.35	48.65	49.00	49.22	49.44	**50.09**	49.80	48.29
Attn only	Cover.	0.894	0.895	0.895	0.895	0.895	0.894	0.893	0.886
	Score	41.29	42.36	44.62	44.91	45.74	**47.52**	46.69	43.35
Dissim only	Cover.	0.903	0.903	0.903	0.904	0.904	0.905	0.905	0.905
	Score	**43.86**	42.51	42.37	43.15	41.85	41.51	41.21	39.62

Table 5: Directional coverage ratio and average LongBench score across all 16 subtasks and three models (Llama-3.1-8B, Qwen2.5-7B, Qwen3.5-9B) under different block sizes and scoring modes at $\rho = 0.2$. "Cover." denotes the directional coverage ratio; "Score" denotes the average LongBench score.

Comparison	Variant	Avg. LongBench
Fusion, 90% comp.	Default multiplicative	38.53
	Additive	35.29
Exponent, 70% comp.	Adaptive Approx ($\alpha = 0.7$)	43.63
	Fixed exponent ($\alpha = 0.9$)	42.78

Table 6: Additional scoring ablations.

yet yields the lowest LongBench scores (39.62–43.86), confirming that pure diversity retains directionally diverse but semantically irrelevant tokens. Second, importance without coverage underperforms: the attn-only mode achieves only moderate coverage (0.886–0.895) and trails the hybrid mode by 2.6–7.0 points across block sizes. Third, the hybrid mode achieves the best observed balance with the best scores (48.29–50.09, peaking at S=4096) while maintaining high coverage (0.889–0.899). Coverage and scores co-degrade at extreme block sizes ($S \geq 8192$), while the hybrid mode remains relatively stable across block sizes, varying within 1.8 points from S=128 to 16384.

5 Conclusion

We present AnchorKV, a training-free and prefill-only KV cache compression framework for long-context LLM inference. AnchorKV addresses the coverage–importance dilemma by combining protected context zones, block-wise temporal budgeting, anchor-based directional diversity, and attention-based importance. Experiments on LongBench across three backbone models show stronger overall performance than existing training-free baselines, while ablations confirm that block-wise budgeting and anchor-based diversity provide complementary gains. The query-subsampled variant further reduces full-query importance-estimation overhead with only modest performance loss.

Limitations

AnchorKV is designed as a training-free KV compression framework, and several directions remain open for further improvement.

Model and Task Adaptivity. AnchorKV uses causal-attention importance as a lightweight context-dependent signal and combines it with anchor-based diversity through a simple hybrid score. This design works well across the evaluated models, but the relative usefulness of importance and diversity may vary across model architectures and task types. More adaptive mechanisms could further adjust the balance between the two signals on a per-model or per-input basis.

Single Default Configuration. For simplicity and reproducibility, AnchorKV uses one default configuration for block size S, protected prefix/recent token counts (L_{prefix}, L_{recent}), and query sampling ratio ρ_q across all experiments. This avoids per-benchmark tuning and makes the comparison cleaner. Model- or task-specific configu-

rations may further improve performance, but we leave such adaptive tuning to future work.

Budget Allocation. AnchorKV adopts uniform per-block budgeting to preserve temporal coverage and avoid context holes. This choice is simple and stable, but it does not explicitly model differences in information density across blocks. Future work may combine the current local-budget constraint with adaptive allocation, while retaining the coverage protection provided by block-wise selection.

Hybrid Scoring Rule. We use the simple retention-adaptive rule $\alpha = 1 - \rho$ for the hybrid scoring exponent. This keeps the method training-free and easy to deploy. A learned or analytically derived rule could potentially provide a finer balance between directional coverage and attention-based importance under different retention ratios.

Evaluation Scope. Our main evaluation focuses on LongBench, which provides a broad testbed for long-context understanding. Extending evaluation to more extreme long-context benchmarks such as RULER (Hsieh et al., 2024) and InfiniteBench (Zhang et al., 2024), as well as incorporating stronger semantic or human evaluation for generation quality, would provide a more complete picture of AnchorKV's behavior under different deployment settings.

Runtime Extension. The experiments in this paper evaluate AnchorKV under a pre-generation compression setting, where the compressed cache is reused during decoding. The same selection principle, however, is not restricted to this setting: block-wise budgeting, anchor-based diversity scoring, and importance-guided filtering can also be applied periodically during decoding or combined with decode-time KV eviction. We leave a full study of such runtime extensions to future work.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive feedback. Mengbo Wang and Dayu Wang conducted this work under the direct supervision of Dr. Zhuochen Fan. This work was supported in part by the Major Key Project of Pengcheng Laboratory under grant No. PCL2026A01, in part by the Open Project of National Engineering Laboratory for Technology of Internet Domain Name under grant No. KF202512, in part by the National Natural Science Foundation of China (NSFC) under grant No. 62402012, in part by the Basic and Frontier Research Project of Pengcheng Laboratory under grant No. 2025QYB004, and in part by Guangdong S&T Programme under grant No. 2024A0101010001.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.

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. 2024. 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)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2025. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3639–3664, Vienna, Austria. Association for Computational Linguistics.

Payman Behnam, Yaosheng Fu, Ritchie Zhao, Po-An Tsai, Zhiding Yu, and Alexey Tumanov. 2025. RocketKV: Accelerating long-context LLM inference via two-stage KV cache compression. *arXiv preprint arXiv:2502.14051*.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Baobao Chang, Junjie Hu, and Wen Xiao. 2025. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. In *COLM 2025*.

Krishna Teja Chitty-Venkata, Jie Ye, Siddhisanket Raskar, Anthony Kougkas, Xian Sun, Murali Emani, Venkatram Vishwanath, and Bogdan Nicolae. 2026. PagedEviction: Structured block-wise KV cache pruning for efficient large language model inference. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 3207–3218. Association for Computational Linguistics.

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

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S. Kevin Zhou. 2024. Ada-KV: Optimizing KV cache eviction by adaptive budget allocation for efficient LLM inference. *arXiv preprint arXiv:2407.11550*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

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

Jie Hu, Shengnan Wang, Yutong He, Ping Gong, Jiawei Yi, Juncheng Zhang, Youhui Bai, Renhai Chen, Gong Zhang, Cheng Li, and Kun Yuan. 2025. CentroidKV: Efficient long-context LLM inference via KV cache clustering. *arXiv preprint arXiv:2506.11418*.

Dongwon Jo, Jiwon Song, Yulhwa Kim, and Jae-Joon Kim. 2026. FastKV: Decoupling of context reduction and KV cache compression for prefill-decoding acceleration. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 32167–32186.

Hao Kang, Qingru Zhang, Souvik Kundu, Geonhwa Jeong, Zaoxing Liu, Tushar Krishna, and Tuo Zhao. 2024. GEAR: An efficient KV cache compression recipe for near-lossless generative inference of LLM. *arXiv preprint arXiv:2403.05527*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th ACM Symposium on Operating Systems Principles*, pages 611–626.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024. SnapKV: LLM knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*.

Xiang Liu, Zhenheng Tang, Peijie Dong, Zeyu Li, Liuyue, Bo Li, Xuming Hu, and Xiaowen Chu. 2025. ChunkKV: Semantic-preserving KV cache compression for efficient long-context LLM inference. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 28728–28778. Curran Associates, Inc.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. 2023. Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time. *arXiv preprint arXiv:2305.17118*.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32332–32344. PMLR.

Da Ma, Lu Chen, Situo Zhang, Yuxun Miao, Su Zhu, Zhi Chen, Hongshen Xu, Hanqi Li, Shuai Fan, Lei Pan, and Kai Yu. 2024. Compressing KV cache for long-context LLM inference with inter-layer attention similarity. *arXiv preprint arXiv:2412.02252*.

Junyoung Park, Dalton Jones, Matthew Morse, Raghavv Goel, Mingu Lee, and Christopher Lott. 2025. KeyDiff: Key similarity-based KV cache eviction for long-context LLM inference in resource-constrained environments. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 5983–6019. Curran Associates, Inc.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents.

Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Beidi Chen, Percy Liang, Christopher Re, Ion Stoica, and Ce Zhang. 2023. FlexGen: High-throughput generative inference of large language models with a single GPU. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 31094–31116. PMLR.

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

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Guangtao Wang, Shubhangi Upasani, Chen Wu, Darshan Gandhi, Jonathan Li, Changran Hu, Bo Li, and Urmish Thakker. 2025. LLMs know what to drop: Self-attention guided KV cache eviction for efficient long-context inference. *arXiv preprint arXiv:2503.08879*.

Jialong Wu, Zhenglin Wang, Linhai Zhang, Yilong Lai, Yulan He, and Deyu Zhou. 2025. SCOPE: Optimizing key-value cache compression in long-context generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10775–10790. Association for Computational Linguistics.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pages 21875–21895.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞Bench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277, Bangkok, Thailand. Association for Computational Linguistics.

Yanqi Zhang, Yuwei Hu, Runyuan Zhao, John CS Lui, and Haibo Chen. 2025. DiffKV: Differentiated memory management for large language models with parallel KV compaction. In *Proceedings of the ACM SIGOPS 31st Symposium on Operating Systems Principles*, pages 431–445.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 34661–34710. Curran Associates, Inc.

Xiabin Zhou, Wenbin Wang, Minyan Zeng, Jiaxian Guo, Xuebo Liu, Li Shen, Min Zhang, and Liang Ding. 2025. DynamicKV: Task-aware adaptive KV cache compression for long context LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 8042–8057, Suzhou, China. Association for Computational Linguistics.

A Theoretical Analysis

In this section, we provide a mathematical analysis of the AnchorKV framework, focusing on query-subsampled importance estimation, block-wise variance reduction, and the overall asymptotic time complexity. The sampling guarantee below concerns the query-subsampled importance estimator, while the block-wise analysis concerns the budget-allocation mechanism rather than global optimality of the complete hybrid selection rule. We denote the input sequence length as L and the KV head dimension as d. The Key and Value matrices are represented as $K, V \in \mathbb{R}^{L\times d}$.

We first analyze the sampling error of the query importance estimation. As described in Section 3.4, AnchorKV computes causal attention via a dedicated $\mathrm{QK}^\top$ pass:

$$A^{\text{causal}} = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d}} + M\right), \tag{12}$$

where M is the causal mask. The causal importance signal is then

$$I_i = \frac{1}{u_i}\sum_{j=i}^{L} A^{\text{causal}}_{j,i}, \tag{13}$$

where $u_i = L - i + 1$ counts the valid causal query positions for key i, and $A^{\text{causal}}_{j,i} \in [0, 1]$. The Approx variant estimates I_i by uniformly sampling a subset of query positions $\Omega \subseteq \{1, \ldots, L\}$ of size $m = |\Omega|$ without replacement and averaging only over sampled positions that can causally observe key i:

$$\tilde{I}_i = \frac{1}{\tilde{u}_i}\sum_{j\in\Omega,\, j\geq i} A^{\text{causal}}_{j,i}, \quad \tilde{u}_i = |\{j \in \Omega : j \geq i\}|. \tag{14}$$

If $\tilde{u}_i = 0$, the implementation assigns zero sampled importance before score normalization. The causal constraint ($j \geq i$) means later tokens are observed by fewer sampled queries; the valid-query normalization above makes this explicit.

Lemma 1 (Conditional Concentration of Valid–Query Importance Estimation)**.** *Fix a key position i and let $\mathcal{Q}_i = \{j : i \leq j \leq L\}$ be the set of $u_i = L - i + 1$ query positions that can causally observe it. The implementation samples a shared global query set $\Omega \subseteq \{1, \ldots, L\}$ of size m uniformly without replacement, and the effective number of valid sampled queries is*

$$\tilde{u}_i = |\Omega \cap \mathcal{Q}_i| \sim \text{Hypergeometric}(L, u_i, m), \tag{15}$$

		Single Doc. QA			Multi Doc. QA			Summarization			Few-shot			Synthetic		Code	
		NarrQA	Qasper	MF-en	HotpotQA	2WikiMQA	Musique	GovRep	QMSum	MultiNews	TREC	TriviaQA	SAMSum	PCount	PR-en	Lcc	RB-P
Qwen3.5-9B		23.82	30.43	35.07	47.70	47.87	28.71	28.44	18.83	22.35	60.00	89.73	30.81	7.50	28.50	16.51	56.63
H2O	0.1	13.56	12.54	18.85	30.55	34.00	16.23	14.85	17.10	8.16	35.00	80.63	30.69	8.00	5.50	17.96	55.48
	0.2	17.88	16.03	24.70	34.04	33.00	19.23	20.89	17.47	12.38	51.75	85.29	30.09	7.50	5.00	13.97	56.41
	0.3	21.19	20.92	28.45	41.40	35.89	22.51	24.90	18.10	16.25	55.25	86.80	30.94	7.50	8.50	13.61	57.32
StreamingLLM	0.1	9.74	9.53	16.14	22.76	27.88	12.25	12.61	15.95	6.85	0.50	78.57	30.46	0.00	6.00	21.03	56.32
	0.2	10.18	10.58	18.57	22.88	30.61	12.70	13.67	16.20	7.25	0.50	79.98	30.47	0.00	5.00	21.05	55.85
	0.3	11.13	10.14	19.57	26.06	33.93	13.88	14.80	16.77	8.37	1.00	79.92	30.65	0.00	5.00	21.17	55.82
SnapKV	0.1	17.68	13.91	20.39	33.84	33.82	18.05	17.17	16.91	8.36	27.75	88.32	31.23	7.00	7.00	19.95	54.19
	0.2	20.14	16.88	24.91	36.88	37.26	21.12	21.87	17.61	11.20	36.75	88.48	31.88	8.00	10.00	18.02	56.14
	0.3	23.57	20.28	27.69	39.99	39.27	22.34	24.49	18.20	14.72	42.25	88.50	32.09	7.50	13.50	15.55	57.15
KeyDiff	0.1	15.59	13.74	16.85	28.99	33.45	13.19	14.48	16.64	8.22	12.50	87.94	30.52	1.00	5.50	19.87	53.95
	0.2	18.60	13.64	18.53	29.84	33.80	14.68	17.77	16.95	8.35	17.00	87.95	30.77	3.50	7.50	18.38	54.23
	0.3	18.51	14.74	20.72	29.60	33.44	18.34	19.83	16.98	8.57	25.00	88.05	30.85	6.00	10.00	16.07	53.74
AnchorKV (Full)	0.1	20.13	17.15	21.99	37.51	36.74	21.35	18.87	17.18	8.95	43.75	87.53	30.40	7.50	6.50	20.84	56.11
	0.2	22.86	19.57	27.06	40.84	37.17	24.43	24.74	18.13	13.22	52.75	87.88	31.19	7.50	6.50	16.26	57.47
	0.3	22.98	21.26	28.97	43.41	41.21	24.81	26.83	18.42	14.43	55.50	88.17	30.88	7.50	10.00	14.23	57.66
AnchorKV (Approx)	0.1	17.80	15.22	19.96	34.91	35.13	20.15	17.86	17.29	8.35	37.25	87.68	30.65	7.50	5.50	18.99	55.85
	0.2	21.86	18.94	25.36	39.41	34.29	23.56	23.53	17.96	11.63	52.25	88.21	31.28	7.50	5.50	14.57	56.78
	0.3	22.34	20.20	27.62	41.95	37.07	23.87	26.10	18.12	15.38	52.25	89.13	31.21	7.50	8.50	13.81	57.80

Table 7: Qwen3.5-9B results on LongBench across all methods and retention ratios (continuation of Table 1).

with $\mathbb{E}[\tilde{u}_i] = mu_i/L$. *Conditioned on* $\tilde{u}_i = s > 0$, *for any* $\epsilon > 0$,

$$\mathbb{P}\Big(|\tilde{I}_i - I_i| \geq \epsilon \mid \tilde{u}_i = s\Big) \leq 2\exp\big(-2s\epsilon^2\big). \tag{16}$$

Proof. Conditioned on $\tilde{u}_i = s > 0$, the set $\Omega \cap \mathcal{Q}_i$ is a uniform sample without replacement of size s from $\mathcal{Q}_i$. Since $A^{\text{causal}}_{j,i} \in [0,1]$, $\tilde{I}_i$ is the empirical mean of a bounded finite population whose mean is I_i. Hoeffding's inequality for sampling without replacement yields the stated conditional bound. □

The probability that no sampled query can causally observe token i is

$$\mathbb{P}(\tilde{u}_i = 0) = \frac{\binom{L-u_i}{m}}{\binom{L}{m}} \leq \exp\Big(-\frac{mu_i}{L}\Big). \tag{17}$$

Since the final 64 tokens are protected, every token subject to compression has $u_i \geq 65$. With the default query sampling ratio $\rho_q = 0.1$, this gives $\mathbb{P}(\tilde{u}_i = 0) \leq e^{-6.5} \approx 0.0015$ for each compressible position, with smaller probabilities for earlier positions. The bound is therefore position-dependent and does not establish uniform estimation error across all token positions.

We also give a robustness interpretation of uniform local budgeting. Let block m contain n_m tokens and receive retention ratio r_m, subject to

$$\sum_{m=1}^{M} n_m r_m = \rho \sum_{m=1}^{M} n_m. \tag{18}$$

Let p_m denote the unknown probability that future generation depends on block m, and let $g(r_m)$ be its preserved utility. Under arbitrary unknown relevance distributions,

$$\min_{p \in \Delta_M} \sum_m p_m g(r_m) = \min_m g(r_m). \tag{19}$$

Since the weighted average of r_m is fixed at ρ, $\min_m r_m \leq \rho$, with equality when $r_1 = \cdots = r_M = \rho$. Thus, uniform allocation maximizes the minimum protection of any temporal region when future relevance is unknown. This statement concerns the budget-allocation component, not global optimality of the complete hybrid token-selection rule.

Next, we establish the variance reduction property of the Block-wise Partitioning mechanism. Suppose we aim to retain a total of R tokens from the compressible zone of length L_{comp}. A naive global sampling strategy selects tokens globally with probability $\rho = R/L_{\text{comp}}$. AnchorKV, instead, partitions the sequence into $B = L_{\text{comp}}/S$ blocks of size S, and retains a fixed budget of k tokens per block (e.g., $k = \lfloor \rho S \rfloor$ for full blocks).

Lemma 2 (Variance Reduction via Block-wise Partitioning)**.** *Let* $Y = \sum_{i=1}^{L_{comp}} \delta_i v_i$ *be the sum of retained values, where* $\delta_i \in \{0,1\}$ *is the indicator of whether token* i *is retained, and* $v_i \in \mathbb{R}$ *is a scalar feature. Let* $\mathrm{Var}_{\mathrm{g}}(Y)$ *be the variance under global independent Bernoulli sampling, and* $\mathrm{Var}_{\mathrm{b}}(Y)$ *be the variance under AnchorKV's block-wise fixed-budget allocation. If local features within each block are non-negatively correlated in the sense that*

$$\sum_{\substack{i \neq j \\ i,j \in \mathcal{B}_m}} v_i v_j \geq 0 \quad \textit{for every block } \mathcal{B}_m, \tag{20}$$

Method	Avg.	NarrQA	Qasper	MF-en	HotpotQA	2WikiMQA	Musique	GovRep	QMSum	MultiNews	TREC	TriviaQA	SAMSum	PCount	PR-en	Lcc	RB-P
AnchorKV (Full)	38.53	27.43	26.74	29.79	48.37	31.48	24.97	30.50	21.97	20.94	59.00	87.91	40.29	7.00	33.17	63.75	63.14
AnchorKV (Approx)	36.98	27.83	25.42	29.12	42.83	26.81	21.46	30.16	22.07	20.86	55.00	86.39	40.52	6.50	30.17	63.75	62.71
AdaKV-SnapKV	34.12	21.03	14.71	25.27	36.90	21.79	16.91	26.46	18.06	18.94	41.50	87.68	40.48	7.50	42.83	64.26	61.54

Table 8: Comparison with AdaKV-SnapKV on Qwen2.5-7B at 90% compression.

then $\mathrm{Var}_{\mathrm{b}}(Y) \leq \mathrm{Var}_{\mathrm{g}}(Y)$*, reducing the variance contribution caused by contiguous selection imbalance.*

Proof. Under global independent sampling, $\delta_i \sim \mathrm{Bernoulli}(\rho)$. The variables δ_i are mutually independent, so the variance is $\mathrm{Var}_{\mathrm{g}}(Y) = \sum_{i=1}^{L_{\mathrm{comp}}} v_i^2 \,\mathrm{Var}(\delta_i) = \rho(1-\rho)\sum_{i=1}^{L_{\mathrm{comp}}} v_i^2$.

Under the block-wise partitioning, the sequence is divided into disjoint subsets $\mathcal{B}_1, \ldots, \mathcal{B}_B$. Within each block $\mathcal{B}_m$, exactly k tokens are retained. Thus, the sum of indicators within a block is deterministic: $\sum_{i\in\mathcal{B}_m} \delta_i = k$. Consequently, the covariance between indicators in the same block is negative: $\mathrm{Cov}(\delta_i, \delta_j) = -\frac{\rho(1-\rho)}{S-1}$ for $i \neq j \in \mathcal{B}_m$. For tokens in different blocks, they remain independent, so $\mathrm{Cov}(\delta_i, \delta_k) = 0$.

The variance under block partitioning is then:

$$
\begin{aligned}
\mathrm{Var}_{\mathrm{b}}(Y) &= \sum_{m=1}^{B} \mathrm{Var}(Y_m), \quad Y_m = \sum_{i\in\mathcal{B}_m} \delta_i v_i, \\
\mathrm{Var}(Y_m) &= P_m + N_m, \\
P_m &= \sum_{i\in\mathcal{B}_m} v_i^2 \mathrm{Var}(\delta_i), \\
N_m &= \sum_{\substack{i\neq j \\ i,j\in\mathcal{B}_m}} v_i v_j \mathrm{Cov}(\delta_i, \delta_j), \\
\mathrm{Var}_{\mathrm{b}}(Y) &= \rho(1-\rho)\sum_{i=1}^{L_{\mathrm{comp}}} v_i^2 - \frac{\rho(1-\rho)}{S-1}\sum_{m=1}^{B} C_m, \\
C_m &= \sum_{\substack{i\neq j \\ i,j\in\mathcal{B}_m}} v_i v_j
\end{aligned}
\tag{21}
$$

Under the non-negative local correlation condition in the lemma statement, each C_m is non-negative. The negative covariance term therefore cannot increase the variance and yields $\mathrm{Var}_{\mathrm{b}}(Y) \leq \mathrm{Var}_{\mathrm{g}}(Y)$. When at least one block has $C_m > 0$, the inequality is strict. □

Building upon the established error bounds, we evaluate the algorithmic efficiency of AnchorKV.

Theorem 3 (Time Complexity of AnchorKV)**.** *The total asymptotic time complexity for compressing a KV cache of length L using AnchorKV in the prefill stage is $\mathcal{O}(Ld + mLd + L\log S)$, where d is the KV head dimension, m is the number of sampled queries, and S is the block size. The Full variant ($\rho_q = 1.0$) uses $m = L$, yielding $\mathcal{O}(L^2 d)$ complexity. With a fixed sampling ratio $\rho_q \in (0, 1)$, the Approx variant sets $m = \rho_q L$, which reduces the constant factor by ρ_q (yielding $\approx \rho_q \cdot L^2 d$ operations) but does* not *change the asymptotic order: complexity remains $\mathcal{O}(L^2 d)$. True linear-time $\mathcal{O}(Ld)$ scaling is achieved only when m is a small* constant *independent of L (e.g., $m = O(1)$), at which point the approximation error bound in Lemma 1 becomes looser and accuracy may degrade. In practice, we operate in the fixed-sampling-ratio regime ($\rho_q = 0.1$ in our experiments), where the approximate variant reduces wall-clock prefill latency by up to 2.7× (see Table 2) purely through constant-factor savings, without asymptotic improvement.*

Proof. Coverage diversity modeling computes block anchors and cosine dissimilarity for L tokens in $\mathcal{O}(Ld)$. Importance estimation multiplies m query vectors by $K^\top \in \mathbb{R}^{d\times L}$, costing $\mathcal{O}(mLd)$, with subsequent Softmax and reduction at $\mathcal{O}(mL)$. Block-wise selection requires $\mathcal{O}(S\log S)$ per block for top-k selection, totaling $\mathcal{O}(L\log S)$ across $B = L/S$ blocks. Summing yields $\mathcal{O}(Ld + mLd + L\log S)$. □

B Qwen3.5-9B Full Results

Appendix B reports the complete Qwen3.5-9B results omitted from Table 1 due to space constraints. As shown in Table 7, AnchorKV follows the same trend as the main results, with Full and Approx variants generally outperforming existing training-free baselines under low and medium retention ratios.

Method	Avg.	Easy	Hard	Long	Medium	Short
AnchorKV (Full)	0.2603	0.2708	0.2694	0.2670	0.1721	0.3222
AnchorKV (Approx)	0.2467	0.2500	0.2412	0.2407	0.1907	0.3111
StreamingLLM	0.2466	0.2292	0.2251	0.2500	0.2233	0.3056
H_2O	0.2370	0.2552	0.2186	0.2407	0.1814	0.2889
KeyDiff	0.2335	0.2500	0.2219	0.2222	0.2233	0.2500
SnapKV	0.2149	0.1927	0.2283	0.2222	0.1814	0.2500

Table 9: Additional LongBench-v2 results.

C Additional Evaluation Results

Recent baseline. We additionally compare with AdaKV-SnapKV, which applies adaptive head-level budget allocation on top of SnapKV, under Qwen2.5-7B at 90% compression. As shown in Table 8, AnchorKV (Full) improves the average score from 34.12 to 38.53. The two methods operate on complementary axes: AdaKV allocates budgets across attention heads, whereas AnchorKV determines which tokens to retain within a given head-level budget.

LongBench-v2. Table 9 reports additional LongBench-v2 results using the same pure-prefill compression protocol and baseline implementations as in our LongBench evaluation. AnchorKV (Full) achieves the highest average score of 0.2603, compared with 0.2466 for the strongest baseline, StreamingLLM. AnchorKV (Approx) obtains 0.2467, comparable to the strongest baseline. The improvement is not uniform across subsets: on the Medium subset, Full scores 0.1721, below StreamingLLM and KeyDiff at 0.2233.