ECHOPRESS: QUERY-AGNOSTIC KV CACHE PRUNING VIA VIRTUAL CONTEXT RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

KV cache pruning reduces long-context inference memory usage by evicting less important key–value pairs. KVzip estimates importance through context reconstruction: prompting a model to repeat the context chunk by chunk. This achieves strong compression quality at the cost of additional forward passes. Learned approximations reduce this cost but require model-specific training. We analyze how KVzip identifies important cached information and show how to approximate its reconstruction scores using information already computed during prefill. These findings motivate EchoPress, a training-free method that approximates reconstruction attention using queries and keys from standard prefill. For each request, it reconstructs only the first chunk to calibrate importance scores for the remaining context. Experiments on LongBench and RULER with Qwen3-8B and Llama-3.1-8B-Instruct show that EchoPress matches KVzip in task accuracy across eviction ratios from 50% to 90%, while reducing compression overhead by 1.7–19.6× and total prefill time by up to 2.9×. Code is available at `https://anonymous.4open.science/r/echo-press/`.

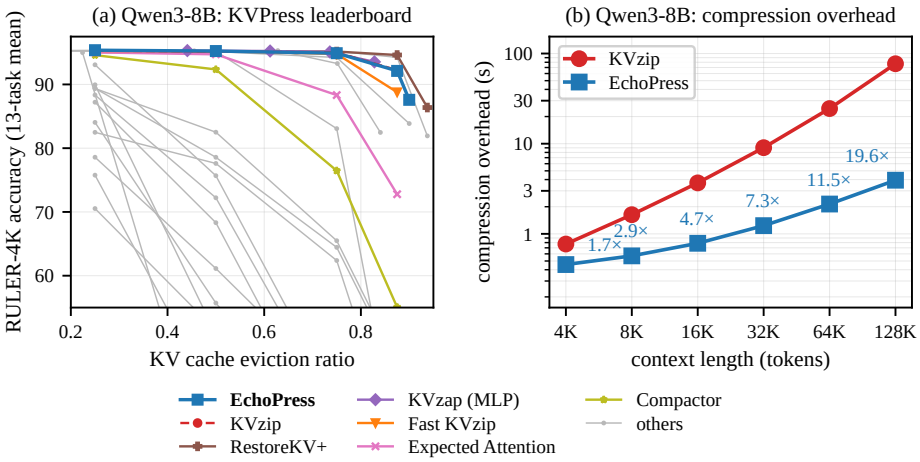

Figure 1: (a) Qwen3-8B RULER-4K accuracy (13-task mean) for query-agnostic methods from the KVPress leaderboard (Hsieh et al., 2024; Jégou et al., 2024), with our EchoPress results overlaid. RestoreKV(+), KVzap, and Fast KVzip use model-specific training (Baek et al., 2026; Jégou & Jeblick, 2026; Kim et al., 2026a). EchoPress requires neither training nor external calibration data. (b) Qwen3-8B compression overhead (additional time beyond plain prefill, including cache compaction) on one 80 GB A100 GPU at 50% eviction. The KVzip baseline uses its official CUDA implementation (Kim et al., 2025).

1 INTRODUCTION

The key–value (KV) cache grows linearly with sequence length, increasing memory use during long-context inference (Kwon et al., 2023). KV eviction reduces this cost by discarding less important KV

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.

pairs. *Query-aware* methods use attention from a question or recent observation window to choose what to retain (Li et al., 2024; Cai et al., 2024), but one compressed cache may not serve a different question reliably (Li et al., 2025). *Query-agnostic* compression uses only the context, allowing the result to be reused across questions, conversation turns, and shared prefixes (Gim et al., 2024; Zheng et al., 2024).

Existing query-agnostic methods trade scoring quality for cost. Prefill-based methods use key norms (Devoto et al., 2024), key geometry (Park et al., 2025; Godey et al., 2025), or expected attention (Devoto et al., 2025) without another model forward pass. Reconstruction methods such as KVzip (Kim et al., 2025) instead prompt the model to repeat the context chunk by chunk after prefill. Each KV pair's importance is scored by the maximum attention it receives during this repetition. A global threshold then evicts the least important pairs across all layers and KV heads. Reconstruction achieves state-of-the-art compression quality but requires expensive additional forward passes over the entire context. In our 4K–128K experiments, KVzip's compression overhead is 2.1–2.2$\times$ the matching plain-prefill time (Section 3.3). Training-based methods replace reconstruction with model-specific score predictors or eviction gates (Jégou & Jeblick, 2026; Kim et al., 2026a), reducing inference cost but requiring a separate training stage.

We ask whether prefill states can approximate reconstruction scores without training a predictor. Reconstruction differs from prefill in three respects: (1) each repetition query can attend to every cached key in its original chunk, including keys that followed it in the original token order, while attention within the repeated copy remains causal; (2) repetition queries are positioned after the cached context, shifting their RoPE coordinates relative to prefill (Su et al., 2024); and (3) reconstruction queries are formed from hidden states conditioned on the full context rather than only a causal prefix. Through these hidden states, context keys outside the target chunk can influence reconstruction queries even when excluded from KVzip's scoring softmax. Our experiments show that exposing all cached keys in the original chunk (difference 1) modestly improves agreement between our prefill-based scores and KVzip's reconstruction scores, while correcting the positional shift (difference 2) gives the largest gain. Together, these two corrections raise the mean per-request median Spearman correlation from 0.208 to 0.678 on Qwen3-8B and from 0.237 to 0.740 on Llama-3.1-8B-Instruct (Section 3.4).

These findings motivate EchoPress's *virtual context reconstruction* (Figure 2). It reuses prefill queries and cached keys, corrects the queries' RoPE positions, and adds a causal virtual-copy branch. Virtual scores preserve useful within-head rankings, but their scales differ across layers and KV heads, complicating global eviction. First-chunk prefill queries also differ most from reconstruction queries because they have seen the least preceding context. EchoPress therefore reconstructs the first chunk exactly and pairs its exact and virtual scores to construct a per-layer, per-KV-head quantile map. The map places later virtual scores on the exact scale, so each request self-calibrates without training. To compute virtual scores efficiently for all later chunks, we implement the scoring operation with fused Triton kernels (Tillet et al., 2019).

Our contributions are:

- An analysis of KVzip's reconstruction scoring that separates key visibility, rotary positions, and context conditioning. It identifies the role of position correction in recovering within-head rankings and of score-scale calibration in global eviction.
- EchoPress, a training-free compression method derived from this analysis. A single exact first-chunk reconstruction supplies request-specific calibration for each layer and KV head, requiring no external calibration data or stored calibration artifacts.
- Evaluation on LongBench and RULER-4K with Qwen3-8B and Llama-3.1-8B-Instruct. EchoPress stays within 0.9 points of KVzip at 50–75% eviction, exceeds it on all four model–benchmark pairs at 90%, and reduces compression overhead by 1.7–19.6$\times$ on an A100 GPU at 4K–128K tokens.

2 ECHOPRESS: VIRTUAL CONTEXT RECONSTRUCTION

EchoPress approximates KVzip's reconstruction scores from queries and keys computed during prefill. It scores every chunk virtually, reconstructs the first chunk exactly, and uses the paired first-chunk scores to calibrate later virtual scores. Figure 2 summarizes the method.

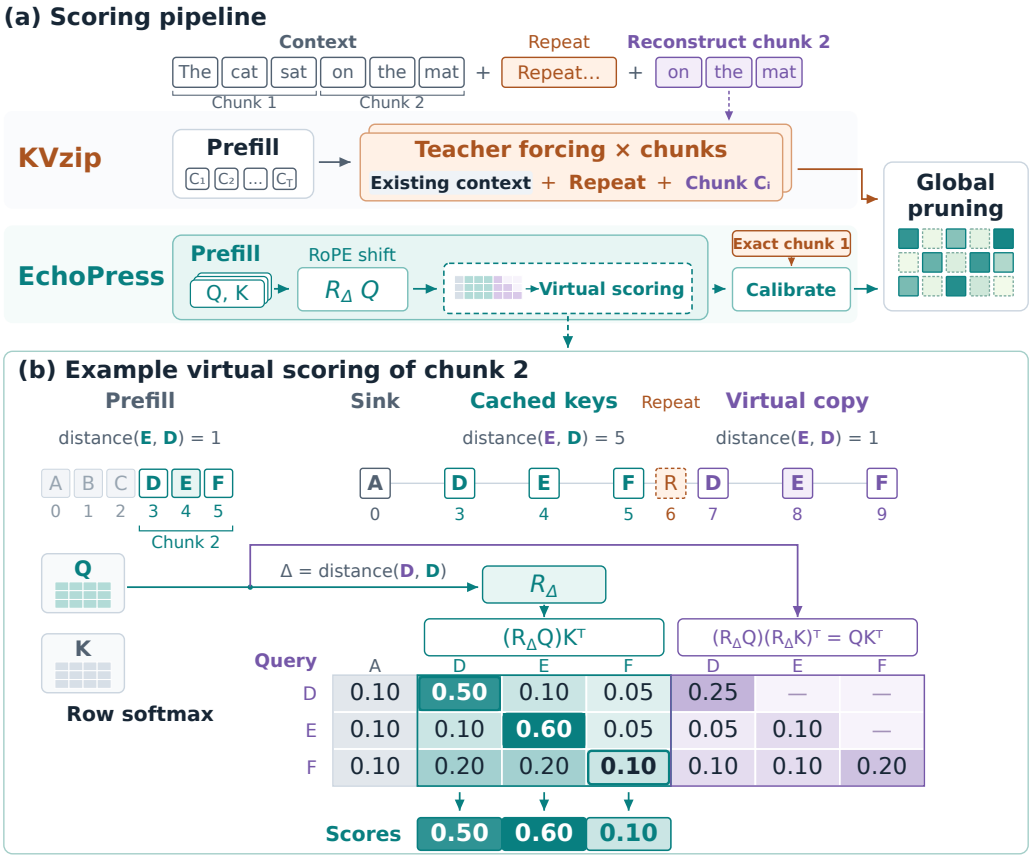

Figure 2: EchoPress scoring pipeline and virtual reconstruction. (a) KVzip reconstructs every chunk. EchoPress reconstructs only the first, reuses prefill states to score later chunks, and calibrates their scores using the first chunk for each layer and KV head before global pruning (dashed cells). (b) Virtual scoring combines ① full visibility of teal cached keys in the target chunk, ② RoPE-shifted queries, and ③ a causal purple virtual copy. Shared rotations cancel in the copy, allowing reuse of $QK^\top$. Prompt R determines the offset but is omitted from scoring. For each query, a joint softmax normalizes attention over sink, cached, and causally visible copy keys. Each cached key's raw score is the largest attention probability it receives across query positions and query heads sharing its KV head. The probabilities shown are illustrative; the full attention matrix and repeated cache are never materialized.

2.1 Reconstruction scoring in KVzip

Consider a decoder with L layers, H query heads, and H_{kv} KV heads of dimension d, with $G = H/H_{kv}$ query heads sharing each KV head (Ainslie et al., 2023). The prefilled cache contains n tokens, partitioned into chunks $\mathcal{C}_t = [a_t, b_t)$ of length $m_t \leq C$. For each chunk, KVzip (Kim et al., 2025) runs a teacher-forced forward pass using the full cache as a prefix. Its input contains a repeat prompt $\boldsymbol{p}_t$ of length P_t (e.g., "Repeat the part of the previous context exactly"), followed by the original chunk. The temporary reconstruction cache is discarded after each chunk. Although this forward pass uses the full original cache, KVzip forms its importance softmax from only the initial sink keys, the target cached chunk, and the causal prefix of the repeated chunk. Other context keys affect the reconstruction hidden states but do not enter the scoring softmax.

We use the scoring rule implemented in KVPress (Jégou et al., 2024). To make the conditioning difference explicit, we reserve unadorned $\boldsymbol{q}, \boldsymbol{k}$ for vectors produced during the original causal prefill and use $\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{k}}$ for vectors recomputed by KVzip after the full original cache has been supplied as a prefix. Thus, suppressing layer and KV-head indices, $\tilde{\boldsymbol{q}}^g_u, \tilde{\boldsymbol{k}}_u$ are full-context-conditioned reconstruction vectors at position $n+u$, whereas $\boldsymbol{k}_i$ is the causally computed key stored for original context position

i. All queries and keys include their rotary embeddings. The score for cached position $i \in \mathcal{C}_t$ is

$$s_i^{\ell h} = \max_{\substack{1 \le g \le G \\ 0 \le u < P_t + m_t}} \frac{\exp(\tilde{\boldsymbol{q}}_u^g \cdot \boldsymbol{k}_i / \sqrt{d})}{Z_u^g}, \tag{1}$$

$$Z_u^g = \sum_{v \in \mathcal{S}_t \cup \mathcal{C}_t} \exp(\tilde{\boldsymbol{q}}_u^g \cdot \boldsymbol{k}_v / \sqrt{d}) + \sum_{v=0}^{u} \exp(\tilde{\boldsymbol{q}}_u^g \cdot \tilde{\boldsymbol{k}}_v / \sqrt{d}),$$

where $\mathcal{S}_t = [0, \min(4, a_t))$ contains initial sink keys. At eviction ratio r, a global threshold removes the $\lfloor rLH_{kv}n \rfloor$ lowest-scoring pairs across layers, heads, and positions.

2.2 Virtual reconstruction from prefill states

At a high level, virtual reconstruction replaces a KVzip reconstruction pass with an attention calculation over states already produced during prefill. For each virtually scored chunk, EchoPress retains the causal prefill representations, shifts them to the RoPE positions where the repeated tokens would appear, and evaluates two key blocks jointly: the original cached chunk and a causal virtual copy.

Let $\boldsymbol{q}_j^g, \boldsymbol{k}_j$ denote the unadorned prefill vectors at original context position $j \in \mathcal{C}_t$. These vectors are conditioned only on the causal prefix through j, unlike the full-context-conditioned $\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{k}}$ in Section 2.1. They also use a different RoPE position: reconstruction shifts token j by $\Delta_t = n + P_t - a_t$. EchoPress retains the causal prefill states and reproduces this positional shift exactly. We use a hat to denote the resulting virtual vectors:

$$\hat{\boldsymbol{q}}_j^g = \boldsymbol{R}_{\Delta_t} \boldsymbol{q}_j^g, \qquad \hat{\boldsymbol{k}}_j = \boldsymbol{R}_{\Delta_t} \boldsymbol{k}_j. \tag{2}$$

The hats therefore do not represent newly computed full-context states. Instead, they are rotated versions of the original prefill vectors. A shared rotation preserves dot products, giving

$$\hat{\boldsymbol{q}}_j^g \cdot \boldsymbol{k}_i = (\boldsymbol{R}_{\Delta_t} \boldsymbol{q}_j^g) \cdot \boldsymbol{k}_i, \qquad \hat{\boldsymbol{q}}_j^g \cdot \hat{\boldsymbol{k}}_i = (\boldsymbol{R}_{\Delta_t} \boldsymbol{q}_j^g) \cdot (\boldsymbol{R}_{\Delta_t} \boldsymbol{k}_i) = \boldsymbol{q}_j^g \cdot \boldsymbol{k}_i. \tag{3}$$

Thus the cached chunk and its virtual copy share one key tensor. The first branch uses shifted queries against cached keys, whereas the second reuses unshifted query–key products for the virtual copy.

Dropping the repeat prompt's query rows and key columns yields

$$\hat{s}_i^{\ell h} = \max_{\substack{1 \le g \le G \\ j \in \mathcal{C}_t}} \frac{\exp(\hat{\boldsymbol{q}}_j^g \cdot \boldsymbol{k}_i / \sqrt{d})}{\hat{Z}_j^g}, \tag{4}$$

$$\hat{Z}_j^g = \sum_{v \in \mathcal{S}_t \cup \mathcal{C}_t} \exp(\hat{\boldsymbol{q}}_j^g \cdot \boldsymbol{k}_v / \sqrt{d}) + \sum_{\substack{v \in \mathcal{C}_t \\ v \le j}} \exp(\boldsymbol{q}_j^g \cdot \boldsymbol{k}_v / \sqrt{d}).$$

Each shifted query sees the whole cached chunk, while its virtual-copy branch remains causal. The repeat prompt determines P_t in (2) but does not need to be processed by the model. Like (1), (4) is chunk-local. The remaining approximation error comes from reusing causal prefill states and omitting the prompt from the scoring softmax.

2.3 Per-request first-chunk calibration

Virtual scores are useful within a head, but their scale distortion varies across layers and KV heads. Since global eviction compares all scores with one threshold, this distortion can assign the wrong cache budget to a layer–KV-head pair. Figure 3 shows this variation: the ratio between exact and virtual scores changes substantially across layers and KV heads. EchoPress therefore uses the request itself to align each layer–KV-head pair's scale.

During prefill, we retain the virtual scores $\hat{s}_i^{\ell h}$ for the first chunk. We then run one exact reconstruction pass for that chunk, producing $s_i^{\ell h}$. For every layer ℓ and KV head h, let $\hat{s}_{(1)}^{\ell h} \le \cdots \le \hat{s}_{(m_1)}^{\ell h}$ and $s_{(1)}^{\ell h} \le \cdots \le s_{(m_1)}^{\ell h}$ be the sorted virtual and exact first-chunk scores. We define the empirical quantile map

$$M_{\ell h}(\hat{s}_{(k)}^{\ell h}) = s_{(k)}^{\ell h}, \qquad k = 1, \ldots, m_1, \tag{5}$$

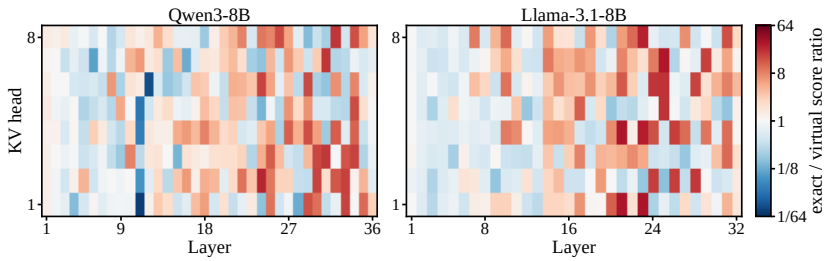

Figure 3: Virtual-score scale distortion across layers and KV heads. For each context and cell, we divide the exact first-chunk score quantile at $q = 0.9$ by the corresponding virtual-score quantile. Each heatmap shows the median ratio over eight LongBench contexts.

with piecewise-linear interpolation between knots and clipping outside the endpoint knots. The exact first-chunk scores are retained. Each later virtual score is replaced by $M_{\ell h}(\hat{s}_i^{\ell h})$.

We use a 2,048-token reconstruction-input budget. The exact first chunk therefore contains at most $2{,}048 - P_1$ context tokens so that it and its repeat prompt fit within this budget. Later chunks can contain up to 2,048 context tokens because virtual scoring does not process the prompt. If the context fits in the exact first chunk, EchoPress coincides with KVzip. Otherwise, it replaces KVzip's remaining reconstruction passes with virtual scoring. Calibration is performed from scratch for every request and uses no external examples, task labels, optimization, or stored model-specific parameters.

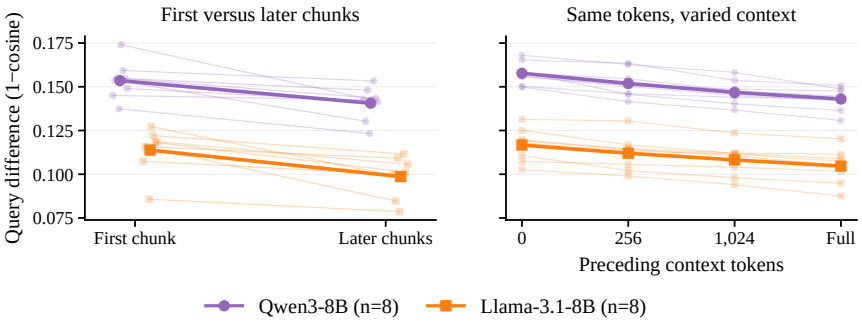

Figure 4: Prefill-to-reconstruction query differences using eight LongBench contexts per model. Thin lines show individual contexts; thick lines show their mean. Left: the first chunk versus the average of later full chunks. Right: fixed second-chunk tokens and positions with varying preceding context, isolating the effect of full-context conditioning. The vertical axis is the mean of one minus the cosine similarity between matched prefill and reconstruction queries, so lower values indicate closer states.

We choose the first chunk for exact reconstruction because its prefill states have seen the least preceding context. Our diagnostic shows that the gap between prefill and reconstruction states is largest for the first chunk and shrinks in later chunks (Figure 4). The exact pass therefore provides both high-quality scores for the first chunk and request-specific calibration data. Appendix C gives measurement details and compares within-chunk score rankings for the first and later chunks.

2.4 Complexity Analysis

We compare the compression work beyond standard prefill, ignoring constant-size repeat prompts and sink sets. For $n \geq C$, the context contains $\Theta(n/C)$ chunks. KVzip reconstructs every chunk against the full n-token cache, costing $O(LHdnC)$ per chunk. EchoPress reconstructs only the first chunk and scores all chunks virtually at $O(LHdC^2)$ per chunk. The additional attention costs are therefore

$$\mathrm{Cost}^{\mathrm{attn}}_{\mathrm{KVzip}} = O(LHdn^2), \tag{6}$$

$$\mathrm{Cost}^{\mathrm{attn}}_{\mathrm{EchoPress}} = O(LHdnC). \tag{7}$$

For fixed C, EchoPress reduces compression attention work from quadratic to linear in n, a factor of n/C. Standard prefill remains quadratic. For model width D, projections and feed-forward layers add $O(LnD^2)$ for KVzip and $O(LCD^2)$ for EchoPress.

Building and applying the calibration maps cost $O(LH_{kv}C \log C)$ and $O(LH_{kv}n \log C)$, respectively. In both the KVPress and KVzip repositories, EchoPress scores each layer during prefill and then releases its queries, retaining only scalar importance scores across layers.

3 EXPERIMENTS

3.1 SETUP

We evaluate Qwen3-8B (Yang et al., 2025) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024), abbreviated Llama-3.1-8B in tables and figures. Accuracy experiments use KVPress (Jégou et al., 2024; Devoto et al., 2025) with bf16, SDPA attention, and batch size one. We compare the uncompressed cache, KVzip, and the EchoPress configuration from Section 2: a 2,048-token reconstruction-input budget, exact first-chunk reconstruction, a separate virtual-to-exact quantile map for each request, layer, and KV head, and global eviction of the lowest-scoring KV pairs. The eviction ratio r is the fraction of KV pairs removed; we test $r \in \{0.5, 0.75, 0.9\}$ on LongBench and RULER-4K.

We evaluate 12 English LongBench tasks (Bai et al., 2024) using KVPress, truncating contexts to 32K tokens. RULER-4K (Hsieh et al., 2024) contains 13 tasks with 500 examples each and uses reference-string matching. Both benchmarks use greedy decoding and per-task output limits. We report equally weighted task averages from one run with seed 42. The evaluator compresses each distinct context once without its questions, then answers its questions independently from that shared cache.

We additionally evaluate RULER at 32K and 64K with 20 examples per task across all 13 tasks, using identical inputs across methods within each model and length. These runs compare the uncompressed cache and both compression methods at $r \in \{0.75, 0.9\}$. The task budgets are 32,768 and 65,536 tokens, including the generator's output allowance. Appendix A.2 details data generation and model settings.

For the latency evaluation, we use KVzip's official CUDA implementation and an EchoPress adapter built on the same codebase, using bf16, FlashAttention-2 (Dao, 2024), batch size one, 2,000-token context chunks, and one 80 GB A100 GPU. Appendix B describes the prefill configurations. Table 10 (Appendix B) summarizes the experimental settings used to evaluate accuracy and timing for KVzip and EchoPress.

3.2 ACCURACY

Figure 5 compares benchmark averages; Tables 2 and 3 give category and task-group means. On LongBench, EchoPress is 0.38 and 0.22 points below KVzip on Qwen at $r = 0.5$ and 0.75, while it is 0.36 and 0.81 points higher on Llama. At $r = 0.9$, EchoPress exceeds KVzip by 1.53 points on Qwen and 1.75 points on Llama.

On RULER-4K, EchoPress and KVzip differ by at most 0.21 points at $r \leq 0.75$. At $r = 0.9$, EchoPress is 0.24 points higher on Qwen and 0.26 points higher on Llama.

At 32K and 64K, EchoPress's mean accuracy ranges from 2.15 points below to 2.72 points above KVzip across both models and eviction ratios (Figure 5, bottom row). At $r = 0.9$, EchoPress is 2.15 points below KVzip on Qwen at 32K and 0.76 points above it at 64K. On Llama, it is 2.53 and 0.15 points higher, respectively. Table 8 gives the mean scores from these 20-example-per-task evaluations.

To compare against a broader set of compression methods, we also run EchoPress at the KVPress leaderboard ratios $r \in \{0.25, 0.5, 0.75, 0.875\}$. Figure 1(a) overlays the Qwen3-8B runs on the public RULER-4K leaderboard snapshot. At $r = 0.875$, EchoPress scores 92.1 on Qwen3-8B and 92.9 on Llama-3.1-8B-Instruct, compared with 92.2 and 92.8 for the leaderboard's published KVzip runs.

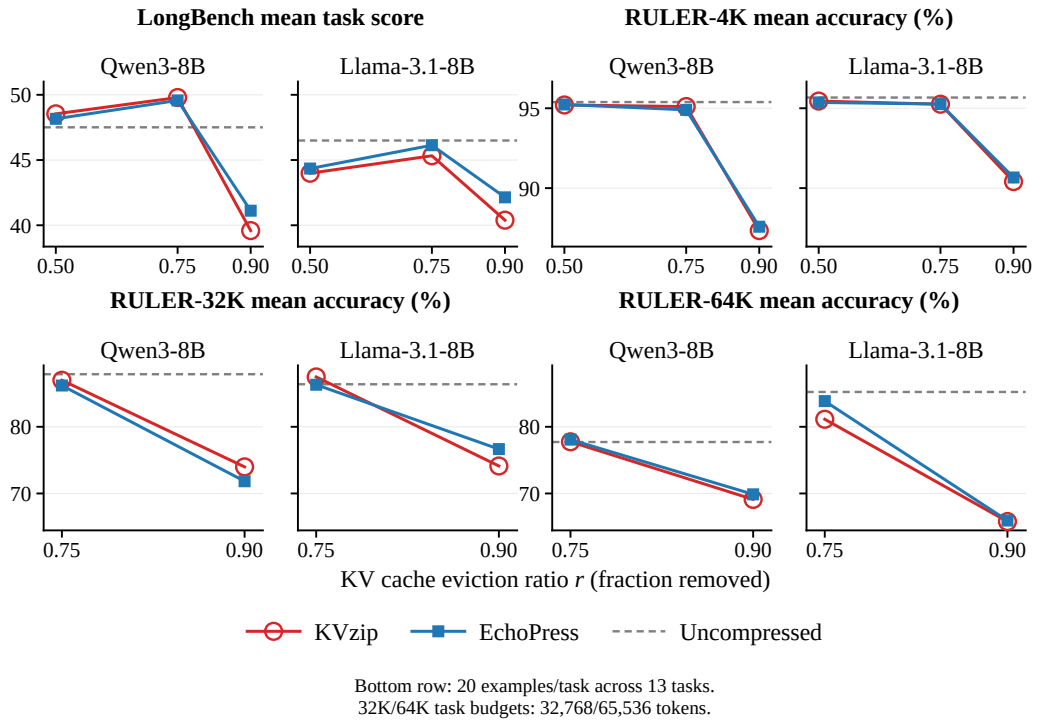

Figure 5: Benchmark averages versus KV eviction ratio r (higher is better). The top row shows LongBench and RULER-4K; the bottom row shows RULER-32K/64K with 20 examples per task. Dashed lines show uncompressed scores. Additional results are in Appendix A.

3.3 Compression latency

Figure 1(b) compares compression overhead for KVzip's official CUDA implementation and our EchoPress adapter at 4K–128K context tokens. We report medians over five repetitions after one warm-up. Compression overhead is the median end-to-end time from the start of prefill through physical compaction minus the median time of the matching plain-prefill configuration. For EchoPress, this measurement includes the exact first-chunk forward, virtual scoring, per-request sorting and interpolation, and compaction. The gap grows with context length because KVzip reconstructs every chunk, whereas EchoPress reconstructs only the first. Appendix B reports the implementation versions and the full compression-overhead sweep (Table 11).

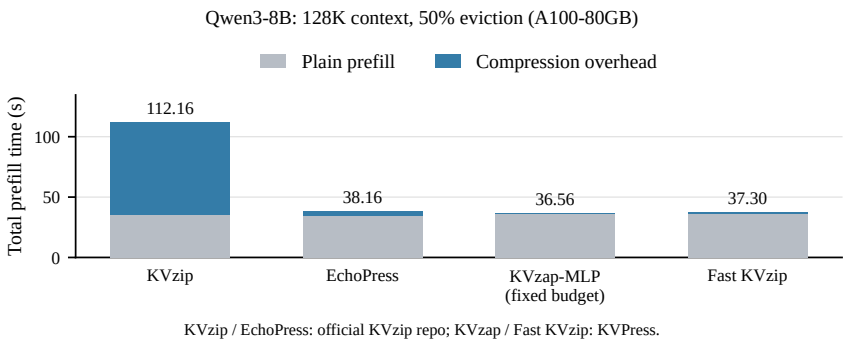

Figure 6: Total prefill time for Qwen3-8B at 128K tokens and a target of 50% eviction on one 80 GB A100 GPU. KVzip and EchoPress use KVzip's official codebase with FlashAttention-2; KVzap-MLP and Fast KVzip use KVPress/SDPA. Each bar stacks its plain-prefill median and compression overhead, including physical compaction. KVzap-MLP uses a fixed retained budget.

At 128K tokens, EchoPress reduces compression overhead from 77.02 to 3.94 seconds on Qwen3-8B, a 19.6× speedup, and from 67.31 to 3.52 seconds on Llama-3.1-8B-Instruct, a 19.1× speedup. Across the complete 4K–128K sweep, speedup grows from 1.7× to 19.6×. Total prefill time (Table 12) falls from 112.16 to 38.16 seconds on Qwen and from 98.26 to 34.02 seconds on Llama, or 2.94× and 2.89×. The stage breakdown in Figure 8 (Appendix B) attributes this gain to scoring and reconstruction.

Figure 6 adds the training-based methods KVzap-MLP and Fast KVzip (Jégou & Jeblick, 2026; Kim et al., 2026a) to this comparison at 128K tokens and a target of 50% eviction. We use their released learned scorers in KVPress, adapting KVzap's threshold policy to a fixed cache budget. Their compression overheads are 0.34 and 1.08 seconds, respectively, against a matching plain-prefill median of 36.22 seconds. EchoPress is training-free and achieves a similar total prefill time in this comparison: 38.16 seconds, compared with 36.56 seconds for KVzap-MLP and 37.30 seconds for Fast KVzip.

The calibration operation itself is inexpensive. In a separate matched KVPress timing ablation, constructing and applying a request-specific map takes at most 1.24 ms longer than applying a precomputed map. This changes 128K compression overhead by less than 1% on both models.

3.4 Analysis and ablations

In Figure 7, we ablate the virtual scorer by adding one component at a time, evaluating on eight LongBench contexts per model. For each context and scorer, we compute one Spearman correlation per layer and KV head across context positions, take the median of those correlations, and then average the eight medians. Exposing all keys in the current cached chunk gives a small improvement over causal prefill attention, raising this statistic from 0.208 to 0.305 on Qwen and from 0.237 to 0.288 on Llama. Applying the RoPE position offset gives the largest gain, reaching 0.678 and 0.740, respectively. Adding the causal virtual-copy branch raises agreement further to 0.720 and 0.768.

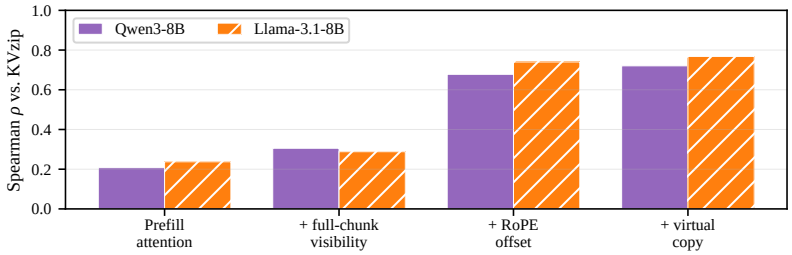

Figure 7: Ablation of the virtual scorer, adding one component at a time. Each bar reports within-head rank agreement (Spearman correlation) with exact reconstruction scores, computed as the mean over eight contexts of the within-context median across layer–KV-head correlations.

Table 1 isolates calibration methods on a fixed 25% subset of RULER-4K at $r = 0.9$. A single global per-request map is insufficient, confirming that scale distortion varies by layer and KV head. Global eviction compares the exact scores retained for the first chunk with virtual scores from all later chunks. We therefore leave the reliable exact scores unchanged and map each later virtual score onto the exact-score scale. Reversing the map instead distorts the exact first-chunk scores by moving them onto the less reliable virtual scale. The proposed per-layer, per-KV-head virtual-to-exact map recovers KVzip accuracy and gives the best needle retrieval score in the table.

4 Related work

KV eviction uses attention scores (Zhang et al., 2023; Oren et al., 2024; Li et al., 2024; Liu et al., 2023a; Chen et al., 2024), layer/head budgets (Cai et al., 2024; Feng et al., 2025; Yang et al., 2024a), or initial and recent tokens (Xiao et al., 2024). FastGen profiles prompts to select head policies; DuoAttention optimizes which heads need full context (Ge et al., 2024; Xiao et al., 2025). FINCH conditions eviction on a prompt; task-aware compression uses task descriptions or examples (Corallo & Papotti, 2024; Corallo et al., 2025). Query-agnostic scores use key geometry or expected attention

Table 1: First-chunk calibration ablation on Qwen3-8B using a fixed 25% RULER-4K subset at $r = 0.9$. Avg. weights all 13 tasks equally. NIAH weights the eight needle-retrieval tasks equally.

Scoring and calibration	RULER Avg.	NIAH (8)
Exact first chunk, no calibration	71.0	72.4
Per-request, per-layer/head exact→virtual	50.8	43.8
Per-request, global virtual→exact	77.3	83.5
EchoPress: per-request, per-layer/head virtual→exact	**87.6**	**95.7**
KVzip	87.5	92.5

(Devoto et al., 2024; Park et al., 2025; Godey et al., 2025; Devoto et al., 2025). Compactor combines non-causal attention and leverage scores (Chari & Van Durme, 2025). InfiniPot uses auxiliary prompts and novelty scores; EpiCache compresses conversational episodes (Kim et al., 2024b; 2026b). LU-KV uses offline head utilities (Tang et al., 2026).

Learned retention includes Locret, DMS, and TRIM-KV (Huang et al., 2024; Łańcucki et al., 2025; Bui et al., 2025). KVzap and Fast KVzip replace KVzip's reconstruction scoring with learned predictors or gates (Jégou & Jeblick, 2026; Kim et al., 2026a). RestoreKV learns cache augmentation before eviction (Baek et al., 2026). EchoPress approximates reconstruction from prefill states and calibrates per request without training. CacheBlend and EPIC inspired our reuse of states under changed context (Yao et al., 2025; Hu et al., 2025).

Other approaches prune key channels (Xu et al., 2025), quantize KV entries (Liu et al., 2024b; Yang et al., 2024b; He et al., 2024), sparsify attention computation (Tang et al., 2024; Jiang et al., 2024; Gao et al., 2024; Lu et al., 2025), or selectively access offloaded caches (Lee et al., 2024; Liu et al., 2024a; Chen et al., 2025). These change entry size or cache access; EchoPress permanently removes entries. Deja Vu studies contextual sparsity; LLM-Microscope examines information held by individual tokens (Liu et al., 2023b; Razzhigaev et al., 2025). Our analysis examines which reconstruction computations can be approximated from prefill states.

5 Discussion and limitations

EchoPress compresses the prefilled context but performs no further eviction during decoding, so the cache grows with generated tokens. Prior work manages this growth through eviction (Zhang et al., 2023; Oren et al., 2024; Anagnostidis et al., 2023; Cai et al., 2025; Du et al., 2025) or learned compressed memory (Kim et al., 2024a). Extending our scorer to completed generation chunks would require buffering query states and choosing a scoring frequency that balances buffering cost, cache savings, and latency. We leave this extension and its evaluation to future work.

Benchmark averages also hide task-level regressions: at $r \geq 0.75$, EchoPress trails KVzip on LongBench code completion on both models, and at $r = 0.9$ on RULER common-word extraction for Qwen (Tables 4–7).

6 Conclusion

We introduced EchoPress, a training-free method for query-agnostic KV cache compression. It reuses prefill states to approximate reconstruction scores and calibrates later scores per layer and KV head using one exact first-chunk reconstruction. On LongBench and RULER-4K with Qwen3-8B and Llama-3.1-8B-Instruct, it closely matches KVzip at moderate eviction and exceeds it at 90%. RULER-32K/64K means stay within 2.8 points of KVzip. At 128K tokens, it reduces compression overhead by 19.1–19.6× on an 80 GB A100 GPU across both models.

AI use statement

We used generative AI tools to polish the manuscript, assist with writing code, and run experiments. We did not use generative AI to develop the research idea. The authors reviewed and verified all AI-generated code and AI-polished text, checking the code for correctness and the text for accuracy

and consistency with the research findings. The authors take responsibility for the final manuscript, code, and experimental results.

Reproducibility statement

Our implementation is available in the anonymous code repository. Section 2 specifies virtual scoring, first-chunk calibration, and the pruning rule. Section 3.1 describes the models, datasets, context truncation, decoding settings, eviction ratios, and random seed. Section 3.4 describes the ablations, and Appendix A reports category and per-task results. Appendix A.2 specifies the RULER-32K/64K generation protocol, model settings, and sample counts. Appendix B documents the timing configurations, hardware, software versions, and measurement procedure, including the differences between the accuracy and timing protocols; Appendix B.1 specifies the implementations and learned scorers used in Figure 6. Appendix C details the first-chunk state measurements and the controlled prefix experiment.

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 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Sotiris Anagnostidis, Dario Pavllo, Luca Biggio, Lorenzo Noci, Aurelien Lucchi, and Thomas Hofmann. Dynamic context pruning for efficient and interpretable autoregressive transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 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, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Ngoc Bui, Shubham Sharma, Simran Lamba, Saumitra Mishra, and Rex Ying. Cache what lasts: Token retention for memory-bounded KV cache in LLMs. *arXiv preprint arXiv:2512.03324*, 2025.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024.

Zefan Cai, Wen Xiao, Hanshi Sun, Cheng Luo, Yikai Zhang, Ke Wan, Yucheng Li, Yeyang Zhou, Li-Wen Chang, Jiuxiang Gu, Zhen Dong, Animashree Anandkumar, Abedelkadir Asi, and Junjie Hu. R-KV: Redundancy-aware KV cache compression for reasoning models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Vivek Chari and Benjamin Van Durme. Compactor: Calibrated query-agnostic KV cache compression with approximate leverage scores. *arXiv preprint arXiv:2507.08143*, 2025.

Yilong Chen, Guoxia Wang, Junyuan Shang, Shiyao Cui, Zhenyu Zhang, Tingwen Liu, Shuohuan Wang, Yu Sun, Dianhai Yu, and Hua Wu. NACL: A general and effective KV cache eviction framework for LLM at inference time. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

Zhuoming Chen, Ranajoy Sadhukhan, Zihao Ye, Yang Zhou, Jianyu Zhang, Niklas Nolte, Yuandong Tian, Matthijs Douze, Leon Bottou, Zhihao Jia, and Beidi Chen. MagicPIG: LSH sampling for efficient LLM generation. In *International Conference on Learning Representations (ICLR)*, 2025.

Giulio Corallo and Paolo Papotti. FINCH: Prompt-guided key-value cache compression for large language models. *Transactions of the Association for Computational Linguistics*, 12, 2024.

Giulio Corallo, Orion Weller, Fabio Petroni, and Paolo Papotti. Beyond RAG: Task-aware KV cache compression for comprehensive knowledge reasoning. *arXiv preprint arXiv:2503.04973*, 2025.

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

Alessio Devoto, Yu Zhao, Simone Scardapane, and Pasquale Minervini. A simple and effective L_2 norm-based strategy for KV cache compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 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.

Wenjie Du, Li Jiang, Keda Tao, Xue Liu, and Huan Wang. Which heads matter for reasoning? RL-guided KV cache compression. *arXiv preprint arXiv:2510.08525*, 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yizhao Gao, Zhichen Zeng, Dayou Du, Shijie Cao, Peiyuan Zhou, Jiaxing Qi, Junjie Lai, Hayden Kwok-Hay So, Ting Cao, Fan Yang, and Mao Yang. SeerAttention: Learning intrinsic sparse attention in your LLMs. *arXiv preprint arXiv:2410.13276*, 2024.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive KV cache compression for LLMs. In *International Conference on Learning Representations (ICLR)*, 2024.

In Gim, Guojun Chen, Seung-seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt Cache: Modular attention reuse for low-latency inference. In *Proceedings of Machine Learning and Systems (MLSys)*, 2024.

Nathan Godey, Alessio Devoto, Yu Zhao, Simone Scardapane, Pasquale Minervini, Éric de la Clergerie, and Benoît Sagot. Q-Filters: Leveraging QK geometry for efficient KV cache compression. *arXiv preprint arXiv:2503.02812*, 2025.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yefei He, Luoming Zhang, Weijia Wu, Jing Liu, Hong Zhou, and Bohan Zhuang. ZipCache: Accurate and efficient KV cache quantization with salient token identification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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

Junhao Hu, Wenrui Huang, Weidong Wang, Haoyi Wang, Tiancheng Hu, Zhang Qin, Hao Feng, Xusheng Chen, Yizhou Shan, and Tao Xie. EPIC: Efficient position-independent caching for serving large language models. In *International Conference on Machine Learning (ICML)*, 2025.

Yuxiang Huang, Binhang Yuan, Xu Han, Chaojun Xiao, and Zhiyuan Liu. Locret: Enhancing eviction in long-context LLM inference with trained retaining heads on consumer-grade devices. *arXiv preprint arXiv:2410.01805*, 2024.

Simon Jégou and Maximilian Jeblick. KVzap: Fast, adaptive, and faithful KV cache pruning. *arXiv preprint arXiv:2601.07891*, 2026.

Simon Jégou, Maximilian Jeblick, Alessio Devoto, and David Austin. KVPress: LLM KV cache compression made easy. `https://github.com/NVIDIA/kvpress`, 2024.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Jang-Hyun Kim, Junyoung Yeom, Sangdoo Yun, and Hyun Oh Song. Compressed context memory for online language model interaction. In *International Conference on Learning Representations (ICLR)*, 2024a.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. InfiniPot: Infinite context processing on memory-constrained LLMs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024b.

Minsoo Kim, Arnav Kundu, Han-Byul Kim, Richa Dixit, and Minsik Cho. EpiCache: Episodic KV cache management for long-term conversation on resource-constrained environments. In *International Conference on Machine Learning (ICML)*, 2026b.

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

Adrian Łańcucki, Konrad Staniszewski, Piotr Nawrot, and Edoardo M. Ponti. Inference-time hyper-scaling with KV cache compression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Wonbeom Lee, Jungi Lee, Junghwan Seo, and Jaewoong Sim. InfiniGen: Efficient generative inference of large language models with dynamic KV cache management. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2024.

Yucheng Li, Huiqiang Jiang, Qianhui Wu, Xufang Luo, Surin Ahn, Chengruidong Zhang, Amir H. Abdi, Dongsheng Li, Jianfeng Gao, Yuqing Yang, and Lili Qiu. SCBench: A KV cache-centric analysis of long-context methods. In *International Conference on Learning Representations (ICLR)*, 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Di Liu, Meng Chen, Baotong Lu, Huiqiang Jiang, Zhenhua Han, Qianxi Zhang, Qi Chen, Chengruidong Zhang, Bailu Ding, Kai Zhang, Chen Chen, Fan Yang, Yuqing Yang, and Lili Qiu. RetrievalAttention: Accelerating long-context LLM inference via vector retrieval. *arXiv preprint arXiv:2409.10516*, 2024a.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023a.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Ré, and Beidi Chen. Deja Vu: Contextual sparsity for efficient LLMs at inference time. In *International Conference on Machine Learning (ICML)*, 2023b.

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. In *International Conference on Machine Learning (ICML)*, 2024b.

Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, Zhiqi Huang, Huan Yuan, Suting Xu, Xinran Xu, Guokun Lai, Yanru Chen, Huabin Zheng, Junjie Yan, Jianlin Su, Yuxin Wu, Neo Y. Zhang, Zhilin Yang, Xinyu Zhou, Mingxing Zhang, and Jiezhong Qiu. MoBA: Mixture of block attention for long-context LLMs. *arXiv preprint arXiv:2502.13189*, 2025.

Matanel Oren, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz. Transformers are multi-state RNNs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Junyoung Park, Dalton Jones, Matthew Morse, Raghavv Goel, Mingu Lee, and Christopher Lott. KeyDiff: Key similarity-based KV cache eviction for long-context LLM inference in resource-constrained environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Anton Razzhigaev, Matvey Mikhalchuk, Temurbek Rahmatullaev, Elizaveta Goncharova, Polina Druzhinina, Ivan Oseledets, and Andrey Kuznetsov. LLM-Microscope: Uncovering the hidden role of punctuation in context memory of transformers. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025.

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

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. QUEST: Query-aware sparsity for efficient long-context LLM inference. In *International Conference on Machine Learning (ICML)*, 2024.

Ziyao Tang, Pengkun Jiao, Xinhang Chen, Wei Liu, Shiyong Li, and Jingjing Chen. Predicting future utility: Global combinatorial optimization for task-agnostic KV cache eviction. *arXiv preprint arXiv:2602.08585*, 2026.

Philippe Tillet, H. T. Kung, and David Cox. Triton: An intermediate language and compiler for tiled neural network computations. In *Proceedings of the ACM SIGPLAN International Workshop on Machine Learning and Programming Languages (MAPL)*, 2019.

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

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 (ICLR)*, 2025.

Yuhui Xu, Zhanming Jie, Hanze Dong, Lei Wang, Xudong Lu, Aojun Zhou, Amrita Saha, Caiming Xiong, and Doyen Sahoo. ThinK: Thinner key cache by query-driven pruning. In *International Conference on Learning Representations (ICLR)*, 2025.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Dongjie Yang, Xiaodong Han, Yan Gao, Yao Hu, Shilin Zhang, and Hai Zhao. PyramidInfer: Pyramid KV cache compression for high-throughput LLM inference. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024a.

June Yong Yang, Byeongwook Kim, Jeongin Bae, Beomseok Kwon, Gunho Park, Eunho Yang, Se Jung Kwon, and Dongsoo Lee. No token left behind: Reliable KV cache compression via importance-aware mixed precision quantization. *arXiv preprint arXiv:2402.18096*, 2024b.

Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. CacheBlend: Fast large language model serving for RAG with cached knowledge fusion. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys)*, 2025.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

A ADDITIONAL ACCURACY RESULTS

Tables 2–7 expand the results of Section 3.2 using the settings in Section 3.1. Here r is the fraction of KV pairs evicted.

Table 2: LongBench category means. Avg. weights all 12 tasks equally. Retr. denotes passage retrieval and Code denotes LCC. Bold marks the better compressed method at each r and displayed precision unless tied.

Qwen3-8B Method	r	Single QA	Multi QA	Summ.	Few-shot	Retr.	Code	Avg.
Uncompressed	0	42.4	49.2	29.1	40.7	91.4	64.5	47.5
KVzip	0.5	42.4	48.4	**29.2**	**46.4**	94.0	**64.7**	**48.5**
EchoPress	0.5	42.4	**48.7**	29.1	44.0	**95.0**	63.4	48.2
KVzip	0.75	41.7	49.0	28.7	56.5	92.0	**63.1**	**49.8**
EchoPress	0.75	**42.1**	**50.0**	**29.0**	**56.6**	**93.6**	54.0	49.6
KVzip	0.9	**37.1**	38.5	26.8	46.9	54.2	**47.0**	39.6
EchoPress	0.9	35.9	**39.3**	**27.1**	**49.8**	**76.5**	37.6	**41.1**
Llama-3.1-8B Method	r	Single QA	Multi QA	Summ.	Few-shot	Retr.	Code	Avg.
Uncompressed	0	43.8	48.0	30.1	35.2	100.0	52.0	46.5
KVzip	0.5	44.4	44.2	**30.1**	25.4	100.0	**50.9**	44.0
EchoPress	0.5	44.4	**45.7**	30.0	**25.9**	100.0	50.2	**44.3**
KVzip	0.75	42.4	**43.3**	**29.8**	39.8	98.5	**49.0**	45.3
EchoPress	0.75	**43.8**	42.8	29.5	**45.8**	**100.0**	43.2	**46.1**
KVzip	0.9	**42.0**	38.7	27.8	42.0	61.5	**41.3**	40.4
EchoPress	0.9	40.2	**42.4**	27.8	**45.2**	**81.5**	30.3	**42.1**

Table 3: RULER-4K group means. NIAH denotes eight needle-retrieval tasks, VT variable tracking, CWE/FWE common/frequent-word extraction, and QA the two question-answering tasks. Avg. weights all 13 tasks equally.

Qwen3-8B Method	r	NIAH (8)	VT	CWE	FWE	QA (2)	Avg.
Uncompressed	0	100.0	100.0	98.9	95.4	72.9	95.4
KVzip	0.5	100.0	100.0	99.0	**95.9**	71.4	95.2
EchoPress	0.5	100.0	100.0	**99.1**	95.7	**71.7**	95.2
KVzip	0.75	**99.9**	100.0	**98.9**	**96.4**	**70.8**	**95.1**
EchoPress	0.75	99.8	100.0	98.0	95.9	70.5	94.9
KVzip	0.9	92.8	100.0	**78.2**	**84.6**	64.9	87.3
EchoPress	0.9	**95.5**	100.0	59.4	84.3	**65.5**	**87.6**
Llama-3.1-8B Method	r	NIAH (8)	VT	CWE	FWE	QA (2)	Avg.
Uncompressed	0	99.9	99.9	99.6	94.8	75.1	95.7
KVzip	0.5	99.9	99.9	99.7	94.1	**73.8**	**95.5**
EchoPress	0.5	99.9	99.9	99.7	**94.4**	73.1	95.4
KVzip	0.75	100.0	**99.9**	**98.7**	**94.5**	72.8	95.3
EchoPress	0.75	100.0	99.8	98.2	93.9	**73.4**	95.3
KVzip	0.9	**98.1**	**99.3**	69.4	89.9	66.0	90.4
EchoPress	0.9	98.0	98.9	**70.9**	89.9	**67.4**	**90.7**

A.1 PER-TASK RESULTS

LongBench uses task-specific metrics and contexts truncated to 32K tokens; RULER-4K uses string-match accuracy on 500 examples per task.

Table 4: LongBench per-task scores on Qwen3-8B.

Question answering Method	r	Qasper	MultiFQA	NrtvQA	HotpotQA	2WikiMQA	MuSiQue
Uncompressed	0	44.8	55.6	26.7	62.9	49.2	35.4
KVzip	0.5	44.5	55.8	**26.8**	62.8	47.8	**34.7**
EchoPress	0.5	**44.6**	**56.4**	26.1	**62.9**	**48.8**	34.5
KVzip	0.75	42.6	**55.0**	27.6	**63.8**	47.8	35.3
EchoPress	0.75	**43.4**	54.4	**28.4**	63.7	**48.9**	**37.3**
KVzip	0.9	**36.9**	**51.0**	23.2	53.3	37.7	24.4
EchoPress	0.9	35.1	48.0	**24.5**	**53.8**	**39.6**	**24.5**

Other tasks Method	r	GovReport	QMSum	TREC	SAMSum	PsgRetr	LCC
Uncompressed	0	33.6	24.5	41.0	40.3	91.4	64.5
KVzip	0.5	**33.8**	**24.7**	**52.5**	40.4	94.0	**64.7**
EchoPress	0.5	33.6	24.6	47.0	**41.0**	**95.0**	63.4
KVzip	0.75	33.4	24.0	72.5	**40.6**	92.0	**63.1**
EchoPress	0.75	**33.7**	**24.3**	**73.0**	40.1	**93.6**	54.0
KVzip	0.9	30.9	22.6	57.0	36.7	54.2	**47.0**
EchoPress	0.9	**31.4**	**22.8**	**60.8**	**38.8**	**76.5**	37.6

Table 5: LongBench per-task scores on Llama-3.1-8B-Instruct.

Question answering Method	r	Qasper	MultiFQA	NrtvQA	HotpotQA	2WikiMQA	MuSiQue
Uncompressed	0	47.1	55.3	28.9	59.5	51.8	32.7
KVzip	0.5	47.3	56.8	**29.1**	55.8	48.2	28.6
EchoPress	0.5	47.3	**57.0**	28.7	**57.4**	**49.7**	**29.9**
KVzip	0.75	44.6	54.8	27.9	**56.1**	**46.4**	27.4
EchoPress	0.75	**46.0**	**55.6**	**29.7**	54.2	46.3	**28.0**
KVzip	0.9	**39.8**	**55.5**	**30.8**	51.7	39.4	25.1
EchoPress	0.9	39.4	53.1	28.0	**55.5**	**46.1**	**25.6**

Other tasks Method	r	GovReport	QMSum	TREC	SAMSum	PsgRetr	LCC
Uncompressed	0	35.3	24.9	29.5	40.9	100.0	52.0
KVzip	0.5	**34.8**	**25.4**	**16.5**	34.3	100.0	**50.9**
EchoPress	0.5	34.7	25.3	15.5	**36.4**	100.0	50.2
KVzip	0.75	**34.4**	**25.1**	47.0	32.6	98.5	**49.0**
EchoPress	0.75	34.1	24.9	**59.0**	32.6	**100.0**	43.2
KVzip	0.9	**31.8**	23.9	45.5	**38.5**	61.5	**41.3**
EchoPress	0.9	31.5	**24.2**	**52.5**	38.0	**81.5**	30.3

Table 6: RULER-4K per-task accuracy on Qwen3-8B. S1–S3 and MK1–MK3 are single-needle and multi-key variants. MQ/MV are multi-query/value retrieval.

Needle retrieval Method	r	S1	S2	S3	MK1	MK2	MK3	MQ	MV
Uncompressed	0	100.0	100.0	100.0	100.0	100.0	100.0	99.9	100.0
KVzip	0.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
EchoPress	0.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
KVzip	0.75	100.0	100.0	100.0	**99.8**	100.0	**99.8**	100.0	99.9
EchoPress	0.75	100.0	100.0	100.0	99.6	100.0	99.2	100.0	**100.0**
KVzip	0.9	100.0	95.4	98.4	83.0	97.2	**92.0**	97.8	79.0
EchoPress	0.9	100.0	**99.0**	**99.0**	**91.6**	**98.0**	90.6	**98.0**	**87.8**

Other tasks Method	r	VT	CWE	FWE	QA1	QA2
Uncompressed	0	100.0	98.9	95.4	82.0	63.8
KVzip	0.5	100.0	99.0	**95.9**	80.6	62.2
EchoPress	0.5	100.0	**99.1**	95.7	80.6	**62.8**
KVzip	0.75	100.0	**98.9**	**96.4**	**81.0**	**60.6**
EchoPress	0.75	100.0	98.0	95.9	80.6	60.4
KVzip	0.9	100.0	**78.2**	**84.6**	**74.4**	55.4
EchoPress	0.9	100.0	59.4	84.3	74.2	**56.8**

Table 7: RULER-4K per-task accuracy on Llama-3.1-8B-Instruct.

Needle retrieval Method	r	S1	S2	S3	MK1	MK2	MK3	MQ	MV
Uncompressed	0	100.0	100.0	99.8	99.8	100.0	99.8	99.9	99.9
KVzip	0.5	99.8	100.0	100.0	100.0	100.0	99.8	100.0	100.0
EchoPress	0.5	99.8	100.0	100.0	100.0	100.0	99.8	100.0	100.0
KVzip	0.75	100.0	100.0	99.8	100.0	100.0	**100.0**	99.9	99.9
EchoPress	0.75	100.0	100.0	**100.0**	100.0	100.0	99.8	**100.0**	**100.0**
KVzip	0.9	100.0	99.8	**100.0**	98.8	**98.6**	95.6	**99.1**	92.8
EchoPress	0.9	100.0	**100.0**	96.4	**99.8**	98.2	**97.0**	98.3	**94.5**

Other tasks Method	r	VT	CWE	FWE	QA1	QA2
Uncompressed	0	99.9	99.6	94.8	87.8	62.4
KVzip	0.5	99.9	99.7	94.1	**85.6**	**62.0**
EchoPress	0.5	99.9	99.7	**94.4**	85.4	60.8
KVzip	0.75	**99.9**	**98.7**	**94.5**	84.8	60.8
EchoPress	0.75	99.8	98.2	93.9	**85.6**	**61.2**
KVzip	0.9	**99.3**	69.4	89.9	77.6	54.4
EchoPress	0.9	98.9	**70.9**	89.9	**78.6**	**56.2**

A.2 RULER AT 32K AND 64K

Table 8 reports equally weighted means over 13 RULER tasks. At each model and length, all five configurations share the same 260 examples (20 per task): uncompressed, KVzip, and EchoPress, with $r \in \{0.75, 0.9\}$ for compression. Table 9 lists the evaluation settings.

Table 8: RULER accuracy at 32K and 64K, using 20 examples per task. Full denotes the uncompressed baseline. Δ is EchoPress minus KVzip in percentage points.

		Qwen3-8B				Llama-3.1-8B			
Budget	r	Full	KVzip	EchoPress	Δ	Full	KVzip	EchoPress	Δ
32K	0.75	87.88	86.98	86.19	−0.79	86.38	87.47	86.31	−1.17
32K	0.9	87.88	73.99	71.83	−2.15	86.38	74.12	76.65	+2.53
64K	0.75	77.72	77.75	78.08	+0.33	85.22	81.13	83.85	+2.72
64K	0.9	77.72	69.11	69.87	+0.76	85.22	65.79	65.94	+0.15

Table 9: Settings for the RULER-32K/64K evaluation.

Setting	Configuration
Data	NVIDIA/RULER v1 (`c3f5e3b4f87f`), base template, seed 42, model-specific tokenizers; no truncation.
Lengths	Budgets of 32,768 / 65,536 tokens include output allowances; actual prefill prefixes span 28,057–32,718 / 60,224–65,478 tokens.
Runtime	bf16, SDPA, batch size one; PyTorch 2.8.0 (CUDA 12.8), transformers 5.2.0, Triton 3.4.0.
Positions	Qwen3: fixed YaRN factor four (Peng et al., 2024), original context 32,768, checkpoint RoPE base. Llama: released RoPE configuration. Both use a maximum position limit of 131,072.
Decoding	Greedy with official per-task output limits; Qwen thinking disabled.
Compression	The 2,048-token accuracy configuration, global KV-pair allocation, and compression before the question. Question and generated tokens remain uncompressed. EchoPress uses exact first-chunk scores and per-request, per-layer/KV-head calibration, without an offline map.

RULER-4K uses the published KVPress dataset with a common Llama tokenizer, so it is a separate evaluation, not a controlled point on this sweep.

B Compression latency

Tables 11 and 12 compare three configurations at 50% eviction in the KVzip repository (commit `5d84729`). The unmodified "16K-chunk" configuration prefills in 16,000-token chunks; "full-prefill" changes only this setting. EchoPress scores virtually during full-context prefill, releasing queries after each layer and retaining importance scores; exact first-chunk reconstruction, calibration, and compaction follow.

Table 10: Accuracy and timing settings for KVzip and EchoPress. Both use bf16 and batch size one; timing uses one 80 GB A100 at 50% eviction.

Setting	Accuracy	Timing
Engine / attention	KVPress / SDPA	KVzip / FlashAttention-2
Chunk budget	First chunk: 2,048 tokens including prompt; later chunks: up to 2,048	2,000 context tokens plus repeat prompt
Prompts / prefix	KVPress chat/prefix settings	KVzip prompts and prefix/sinks
Context coverage	RULER 4K/32K/64K; LongBench $\leq$32K	4K–128K, excluding protected prefix
Output	Task scores	Compacted cache; no generation

Timing uses FlashAttention-2 2.7.4.post1, PyTorch 2.3.0 (CUDA 12.1), transformers 4.51.3, and Triton 3.4.0, with Qwen3 YaRN factor four. Each model uses the same saved LongBench inputs across methods and five repetitions after warm-up. CUDA is synchronized at stage boundaries. Prompt preparation, calibration, and compaction are included; loading, tokenization, compilation, warm-up, verification, and cleanup are excluded. A 4K check confirms finite scores/logits and bitwise-identical exact first-chunk scores, without testing downstream accuracy equivalence to the accuracy configuration (Section 2.3).

Total prefill time is the median per-run duration through physical compaction, excluding first-token generation. Compression overhead subtracts each configuration's matching plain-prefill median. The separate KVzip and EchoPress sweeps retain separate full-prefill references. Total medians use raw repetitions, not sums of independently computed stage medians.

Table 11: Plain-prefill baselines and compression overhead in seconds. Speedup is KVzip 16K-chunk overhead divided by EchoPress overhead. "Full K" and "Full E" are the respective sweeps' full-prefill baselines.

Qwen3-8B

Context	Plain 16K-chunk	Plain Full K	Plain Full E	KVzip 16K-chunk	KVzip full prefill	EchoPress	Speedup
4K	0.36	0.36	0.36	0.77	0.78	0.46	**1.7**×
8K	0.73	0.73	0.72	1.63	1.62	0.57	**2.9**×
16K	1.67	1.63	1.61	3.68	3.68	0.79	**4.7**×
32K	4.11	3.96	3.91	9.03	9.04	1.23	**7.3**×
64K	11.23	10.90	10.78	24.53	24.47	2.14	**11.5**×
128K	35.14	34.52	34.23	77.02	77.00	3.94	**19.6**×

Llama-3.1-8B

Context	Plain 16K-chunk	Plain Full K	Plain Full E	KVzip 16K-chunk	KVzip full prefill	EchoPress	Speedup
4K	0.32	0.32	0.32	0.68	0.69	0.41	**1.7**×
8K	0.66	0.66	0.66	1.43	1.43	0.51	**2.8**×
16K	1.50	1.46	1.46	3.23	3.23	0.70	**4.6**×
32K	3.67	3.55	3.54	7.93	7.92	1.09	**7.2**×
64K	9.96	9.68	9.70	21.65	21.67	1.87	**11.6**×
128K	30.95	30.36	30.50	67.31	67.31	3.52	**19.1**×

Table 12: Median total prefill time (seconds), including compression and physical compaction, across five repetitions.

	Qwen3-8B			Llama-3.1-8B		
Context	KVzip 16K-chunk	KVzip full	EchoPress	KVzip 16K-chunk	KVzip full	EchoPress
4K	1.13	1.14	0.81	1.01	1.01	0.74
8K	2.36	2.36	1.29	2.09	2.09	1.17
16K	5.34	5.31	2.39	4.73	4.69	2.17
32K	13.14	12.99	5.15	11.60	11.47	4.64
64K	35.76	35.37	12.92	31.61	31.35	11.57
128K	112.16	111.51	38.16	98.26	97.67	34.02

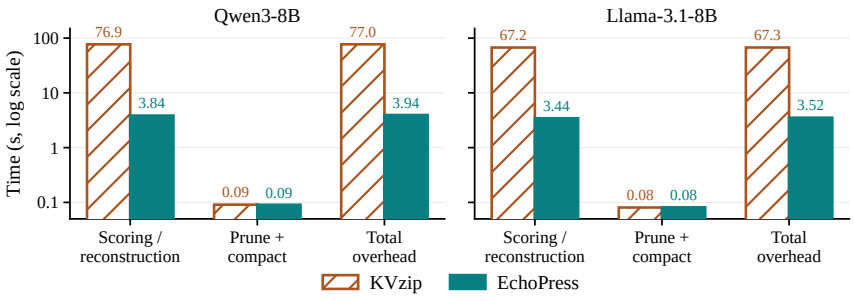

Figure 8: Compression-time breakdown at 128K on one 80 GB A100. For EchoPress, scoring includes exact first-chunk reconstruction, virtual scoring, and calibration. Overhead subtracts the matching plain-prefill baseline.

B.1 TIMING COMPARISON WITH LEARNED METHODS

Figure 6 combines the KVzip 16K-chunk and EchoPress timings above with KVPress/SDPA timings of KVzap-MLP and Fast KVzip (Table 13). All use Qwen3-8B, the same 131,072 context tokens, batch size one, and a 50% eviction target on an A100-SXM4-80GB. Separate sweeps use matching plain-prefill baselines. The KVzip/EchoPress runs use a 28-token protected prefix and retain 0.49973/0.49992 of KV pairs; the learned methods retain exactly 0.5. Backends, prompts, chunking, and compaction differ, so the figure compares measured implementations. Training costs and retained-cache accuracy are not evaluated.

Table 13: Settings for the KVPress implementations of the learned methods.

Setting	Configuration
Input / prefill	Saved concatenated LongBench `gov_report` contexts; three-token chat prefix; 131,075 total tokens; one full-context prefill.
Software	PyTorch 2.8.0 (CUDA 12.8), transformers 5.2.0, Triton 3.4.0; SDPA.
RoPE / precision	Fixed YaRN factor four, original size 32,768, original RoPE base. bf16 model/KVzap weights; released mixed bf16/fp32 Fast KVzip gates.
KVzap-MLP	`nvidia/KVzap-mlp-Qwen3-8B`, revision `bd5c59178466`. Replace fixed threshold with global fixed-budget selection; protect the final 128 tokens.
Fast KVzip	`Jang-Hyun/Fast-KVzip`, revision `67ce1265f510`, `q4_dim16_sink16.pt` gates. Global allocation; promote first four and last 4,096 token scores. KVPress full prefill replaces the released implementation's eviction between prefill chunks.
Repetitions	Five seeded, interleaved repetitions after one warm-up per method; plain-prefill baseline in each repetition; CUDA synchronized at both boundaries.
Timed region	Per-request setup, scoring, selection, physical compaction. Exclude loading, tokenization, compilation/warm-up, validation, and generation.
Packed cache	Head-major, chronological K/V pairs with token positions and per-head offsets; untimed checks of counts, finite tensors, ordered positions, and compactor attention outputs.

C FIRST-CHUNK STATE MEASUREMENTS

Figure 4 uses eight LongBench contexts per model (bf16, SDPA), Section 2's prompts/chunks, and full-cache, unpruned reconstruction. Query differences are $1 - \cos$ after normalization and before RoPE, averaged over 256 evenly spaced token offsets shared by first/later full chunks, query heads, and layers 1 onward; layer 0 checks alignment.

The prefix control fixes second-chunk tokens, positions, and reconstruction reference. Fresh prefill retains the chat prefix, target chunk, and last $b \in \{0, 256, 1024, \text{Full}\}$ preceding tokens; Full is 2,032/2,036 tokens for Qwen/Llama. This isolates context availability; natural chunks also differ in text and repeat prompts.

Table 14 aggregates finite per-layer/KV-head Spearman correlations over positions within each chunk. Section 3.4 instead correlates entire-context scores and takes the median across layers and heads. These diagnostics motivate reconstructing the region with greatest query mismatch, without establishing an optimal calibration chunk.

Table 14: First-chunk diagnostics, equally weighting contexts after averaging later full chunks within each context; partial final chunks are excluded. Arrows compare first with later chunks, except the prefix control ($0 \rightarrow$ Full). The median row replaces only the within-chunk mean over layers and KV heads.

Measurement	Qwen3-8B	Llama-3.1-8B
Query difference, first $\rightarrow$ later	$0.154 \rightarrow 0.141$	$0.114 \rightarrow 0.099$
Query difference, prefix $0 \rightarrow$ Full	$0.158 \rightarrow 0.143$	$0.117 \rightarrow 0.105$
Mean within-chunk Spearman ρ	$0.678 \rightarrow 0.692$	$0.735 \rightarrow 0.745$
Median within-chunk Spearman ρ	$0.717 \rightarrow 0.718$	$0.772 \rightarrow 0.770$