THE KV-SURGEON: COMPENSATION-AWARE OFFLINE KV-CACHE COMPACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline compaction of the key–value cache (KV-cache) reduces the cost of repeatedly querying a fixed context by constructing a smaller cache once and reusing it subsequently. The central challenge of compaction is deciding which cache entries to delete and which to retain. Existing methods score each candidate for deletion on its own, with the rest of the cache being held fixed. However, softmax renormalization of the attention operation couples these decisions in a nonlinear manner: removing one entry changes the attention weights of every remaining one. In this work, we introduce KV-SURGEON, which uses calibration data to take interactions into account in a tractable way. We derive a quadratic surrogate that admits closed-form expressions for jointly making the optimal greedy removal decision as well as compensation within the remaining entries. Across benchmark datasets, such as QuALITY, LongHealth, and Long Code Arena, and at every evaluated compaction ratio, KV-SURGEON substantially advances the cost–accuracy Pareto frontier of dense and mixture-of-experts models from 4B to 30B parameters. We observe gains of up to four accuracy points on question-answering at $50\times$ compaction, exceeding the state of the art, attention matching (Zweiger et al., 2026), at comparable cost. Lastly, we introduce an even faster variant, which also exceeds the state of the art while requiring only two fifths of its cost.

1 INTRODUCTION

Many large language model (LLM) applications repeatedly query a fixed context such as a document, patient record, or codebase. The key–value cache (KV-cache) for such a context grows linearly with its length and occupies a large fraction of the memory used during long-context inference. Every attention layer reads this cache for every generated token, so every request pays for its size in both memory and latency. Offline compaction reduces this cost by processing the context once and compressing the resulting cache, which is then used to serve all later requests (Eyuboglu et al., 2026; Kim et al., 2026; Zweiger et al., 2026).

Unlike serving-time eviction, which discards cache entries within the forward pass of a request (Zhang et al., 2023; Li et al., 2024), offline compaction can spend a larger one-time budget that is amortized over future requests. In particular, it can use *calibration queries*, i.e., query vectors the model produces while re-reading or discussing the context, to decide which entries to retain. Eyuboglu et al. (2026) introduce this regime with *Cartridges*, which trains a new latent cache per context. More recent methods make offline compaction much cheaper (Zweiger et al., 2026).

The central problem in cache compaction is deciding which cache entries can be removed while preserving the resulting attention outputs. For one attention head, a cache of T entries contains a key $k_j \in \mathbb{R}^{d_k}$ and a value $v_j \in \mathbb{R}^{d_v}$ at each position $j = 1, \ldots, T$. A query $q \in \mathbb{R}^{d_k}$ produces

$$p \triangleq \mathrm{softmax}\left(\left(\frac{q^\top k_\ell}{\sqrt{d_k}}\right)_{\ell=1}^{T}\right) \in \mathbb{R}^T, \qquad o \triangleq \sum_{j=1}^{T} p_j v_j \in \mathbb{R}^{d_v},$$

where p_j is the attention weight of entry j and o is the resulting output. Removing entry j does not simply subtract $p_j v_j$, since softmax renormalization changes the attention weights of all remaining entries. The effect of one deletion therefore depends on which other entries remain, making selection

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

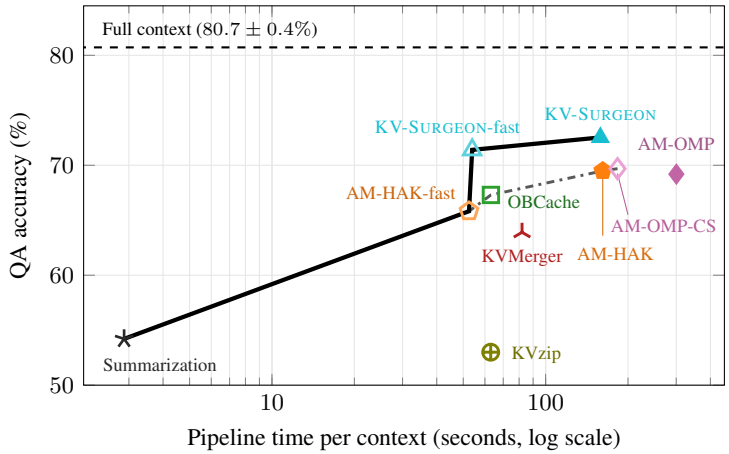


Figure 1: Cost–accuracy frontier at $50\times$ KV-cache compaction on QuALITY with Qwen3-30B-A3B. KV-SURGEON reaches 72.5% accuracy in 159 seconds and KV-SURGEON-fast reaches 71.4% in 54 seconds. Pipeline time per context includes calibration-query generation and compaction. OBCache, KVMerger, and KVzip are adapted to our pipeline (Appendix F). Points are five-seed means. The solid and dash-dotted curves are the Pareto frontiers with and without our methods.

a combinatorial problem rather than a ranking problem. Attention-based scores such as the attention matching methods proposed by Zweiger et al. (2026) additionally overlook value redundancy: if an entry's value is close to the output the head already produces, i.e., $\|v_j - o\|_2^2$ is small, removing it might change the output little, even when its attention weight is large. Good selection must therefore account not only for the contribution of each entry, but also for how deletions interact.

Recent output-aware methods instead estimate the output error caused by deleting an entry (Goel et al., 2026; An et al., 2026; Gu et al., 2026; Feng et al., 2026). CAOTE (Goel et al., 2026) derives the exact change caused by deleting one entry for a single calibration query, and OBCache (Gu et al., 2026) uses a Taylor approximation of the attention output error aggregated across queries. These methods nevertheless evaluate the deletion of one entry while the rest of the cache is held fixed. Attention matching (Zweiger et al., 2026) refits the retained cache to minimize the error of the compacted output, but only after the retained set has been chosen. What is missing is *compensation-aware selection*: each candidate deletion should be scored by the error remaining after the surviving entries have been adjusted to compensate for it, so that selection and refitting are optimized *jointly* rather than in sequence. This is difficult under softmax attention because deletion changes the normalization itself, so the error after both deletion and compensation has no obvious closed form.

In this work, we substantially advance the cost–accuracy Pareto frontier for offline KV-cache compaction. We derive a quadratic surrogate that makes compensation-aware selection tractable. We introduce KV-SURGEON, which uses this surrogate for greedy, compensation-aware elimination. For each candidate deletion, the surrogate admits closed-form expressions for both the optimal compensating update and the resulting increase in surrogate loss. Each candidate entry is assigned a score after the remaining entries have been adjusted to compensate for its removal, and the update is then applied to the cache, so the next deletion is chosen against the compensated entries, as Optimal Brain Surgeon (OBS) (Hassibi & Stork, 1993) does for network weights.

We evaluate KV-SURGEON on QuALITY, LongHealth, and Long Code Arena, using dense and mixture-of-experts models from 4B to 30B parameters and compaction ratios from $5\times$ to $100\times$. Across these settings, KV-SURGEON improves the cost–accuracy frontier at every ratio, achieving higher mean accuracy at comparable cost. Mean question-answering accuracy across our experiments exceeds the state of the art by 4.3 percentage points at $100\times$ compaction and 3.0 percentage points at $50\times$. The gains are less pronounced at $5\times$ and $10\times$, where the strongest methods already suffer almost no deterioration relative to the uncompacted cache. At $50\times$ compaction the state of the art, attention matching with its AM-HAK selection rule (Zweiger et al., 2026), reaches

62.0% accuracy in 138 seconds[1] per context. KV-SURGEON reaches 65.1% in 145 seconds, and KV-SURGEON-FAST reaches 63.9% in 54 seconds. Figure 1 shows the resulting frontier at $50\times$ compaction on the largest model we evaluate.

Our contributions are the following.

1. *Quadratic surrogate.* We derive a quadratic surrogate to the exact attention-output error, which allows us to design efficient methods for approximately solving the compaction problem.
2. *Compensation-aware selection.* Based on this surrogate, we introduce KV-SURGEON, which iteratively removes entries while adjusting the remaining entries in closed form after each deletion, then jointly refits the retained cache. Unlike prior output-aware scores or post-selection refitting, compensation directly influences which entries are retained.
3. *A new cost–accuracy Pareto frontier.* Across QuALITY, LongHealth, and Long Code Arena, dense and mixture-of-experts models from 4B to 30B parameters, and $5\times$ to $100\times$ compaction, KV-SURGEON substantially advances the frontier, with gains at every evaluated ratio and the largest improvements at $50\times$ and $100\times$. We show empirically that compensation during selection is the source of the improvement over exact single-deletion scoring.

2 RELATED WORK

Other forms of KV-cache compression. KV-caches can be compressed along several axes. Quantization lowers the precision of stored keys and values (Liu et al., 2024; Hooper et al., 2024), and low-rank methods reduce the feature dimension (Saxena et al., 2024; Chang et al., 2025). Both are complementary to reducing the number of entries. One common approach, *LLM summarization*, shortens the text input itself by prompting an LLM to summarize it. This approach transfers across models, but gives no direct control over cache size. We study compaction along the sequence dimension, where a fixed budget of entries is retained from an already computed cache.

Offline compaction. Offline methods exploit contexts that are processed once and queried repeatedly. Cartridges (Eyuboglu et al., 2026) train a new latent cache per context from self-generated data. KVzip (Kim et al., 2026) removes cache entries based on the attention they receive while the model repeats the context verbatim. Attention matching (Zweiger et al., 2026) also keeps a subset of the original keys, then fits a scalar logit bias per retained key and refits the stored values to reproduce the original attention outputs on reference queries. Its AM-HAK variant selects entries by root-mean-square attention, and AM-OMP greedily selects keys so that their reweighted attention reproduces the attention mass of the full cache. In both, the support is chosen from attention statistics before the values are refit, so the ability of the retained entries to reproduce the output does not influence selection.

Eviction by proxy scores. Serving-time eviction discards entries within a single request, ranking each one by a score that ignores the rest of the cache: accumulated attention, as in H2O (Zhang et al., 2023) and related methods (Oren et al., 2024; Li et al., 2024; Liu et al., 2023), or a query-free proxy such as position (Xiao et al., 2024), key norm (Devoto et al., 2024), alignment with a principal query direction (Godey et al., 2025), deviation from neighboring keys (Liang et al., 2025), or value norm (Guo et al., 2024). Merging methods soften the same decision by folding evicted entries into retained ones. KVMerger (Wang et al., 2024) merges similar entries into attention-weighted pivots, and Zhang et al. (2024); Wan et al. (2024); Nawrot et al. (2024) use related rules. In all of these, the score of an entry is fixed before any other entry is removed.

Output-aware eviction. More recent methods score entries directly through the attention-output error induced by their removal (Goel et al., 2026; An et al., 2026; Gu et al., 2026; Feng et al., 2026). CAOTE (Goel et al., 2026) and ReST-KV (An et al., 2026) derive the exact effect of deleting one entry for one query. OBCache (Gu et al., 2026) derives a second-order approximation in the style of Optimal Brain Damage (LeCun et al., 1989). CriticalKV (Feng et al., 2026) considers deletion of an arbitrary set for one query and derives an additive upper bound on the resulting output error for

[1] We average computation time for different models and benchmarks as the geometric mean.

selection. These methods differ in how they estimate deletion error, but share a common restriction: candidates are evaluated with the retained cache held fixed.

Our work starts from the same output-reconstruction objective as CAOTE, ReST-KV, and OBCache but changes how it is used for selection. Instead of scoring one deletion against a fixed cache, we derive a quadratic surrogate of the output error over arbitrary deletions and reweightings of the retained entries, aggregated over many calibration queries, and let the retained entries compensate during greedy deletion of cache entries.

3 KV-SURGEON

We now introduce KV-SURGEON, an offline KV-cache compaction method. We first formalize the compaction problem (Section 3.1), then relax the exact output error caused by deleting cache entries and reweighting and modifying the remaining entries into a quadratic surrogate (Section 3.2). KV-SURGEON then compacts each attention head's cache by greedily deleting entries under this surrogate, compensating the survivors after every deletion (Section 3.3), and refitting the retained cache once at the end (Section 3.4).

3.1 PROBLEM FORMULATION

We consider one layer and one KV head with T cache entries and a budget of $t < T$ entries, which may differ between heads (Section 4.1). Under grouped-query attention, several query heads share a KV head, and we pool their queries. Let $K \in \mathbb{R}^{T\times d_k}$ and $V \in \mathbb{R}^{T\times d_v}$ contain the keys and values of the cache, and stack N calibration queries as $Q \in \mathbb{R}^{N\times d_k}$ (cf. Section 4.1). Then, we define the logits $Z \in \mathbb{R}^{N\times T}$ and outputs $O \in \mathbb{R}^{N\times d_v}$ as

$$Z \triangleq QK^\top/\sqrt{d_k}, \quad \text{and} \quad O \triangleq PV,$$

where $P \in \mathbb{R}^{N\times T}$ is the row-wise softmax of Z. We write P_{ij} for the attention paid by query i to cache position j, v_j for row j of V, and o_i for row i of O.

Let $S \subseteq \{1,\dots,T\}$ with $|S| = t$ be the set of retained entries. The compacted cache keeps the original keys, $\widehat{K} \in \mathbb{R}^{t\times d_k}$, the rows of K indexed by S, but may store modified values $\widehat{V} \in \mathbb{R}^{t\times d_v}$. It also carries one weight $w_j > 0$ per retained entry. These weights let the retained entries redistribute attention among themselves, and they control how much attention the compacted block draws in total, which matters when the compacted cache is used in combination with uncompacted tokens such as a system prompt (Appendix G). For query i the compacted cache outputs

$$\widehat{o}_i = \frac{\sum_{j\in S} w_j e^{z_{ij}}\widehat{v}_j}{\sum_{j\in S} w_j e^{z_{ij}}} = \frac{P_{i,S}\,\mathrm{diag}(w)\widehat{V}}{\mu_i}, \qquad \text{with} \quad \mu_i \triangleq P_{i,S}w = \sum_{j\in S} w_j P_{ij}, \tag{1}$$

where $P_{i,S} \in \mathbb{R}^{1\times t}$ is row i of P restricted to the columns indexed by S and μ_i is the attention mass of the compacted cache. The error on the calibration queries is

$$\mathcal{L}(S, w, \widehat{V}) \triangleq \sum_{i=1}^{N} \left\| \frac{P_{i,S}\,\mathrm{diag}(w)\widehat{V}}{P_{i,S}w} - o_i \right\|_2^2. \tag{2}$$

3.2 QUADRATIC SURROGATE

Minimizing (2) is hard in two ways: the choice of S is combinatorial, and for a fixed support the normalization μ_i couples w and $\widehat{V}$. We deal with the second first. Substituting $U \triangleq \mathrm{diag}(w)\widehat{V} \in \mathbb{R}^{t\times d_v}$, which is a bijection because $w > 0$, and splitting the residual gives

$$\mathcal{L}(S, w, U) = \sum_{i=1}^{N} \frac{1}{\mu_i^2}\left\|P_{i,S}U - \mu_i o_i\right\|_2^2 = \sum_{i=1}^{N} \frac{1}{\mu_i^2}\Big\| \underbrace{P_{i,S}U - o_i}_{\text{output reconstruction}} + \underbrace{(1-\mu_i)o_i}_{\text{mass error}} \Big\|_2^2. \tag{3}$$

Inside the norm the residual splits into two parts: the output reconstruction, which measures how well the compacted block reproduces its own output and depends only on U, and the mass error, which measures how much attention mass the block has lost and depends only on w.

We relax the objective in three ways. First, we freeze the normalization by setting $1/\mu_i^2 = 1$. This is exact whenever the compacted block retains the attention mass of the original one, and only the prefactor is frozen: the deviation $1 - \mu_i$ stays inside the norm, so the objective still penalizes any loss of mass. Second, we drop the cross term $2\langle P_{i,S}U - o_i, (1-\mu_i)o_i\rangle$, which leaves the two error sources additive. Third, the squared mass error $(1-\mu_i)^2$ then carries the weight $\|o_i\|_2^2$, and we replace $\|o_i\|_2$ by the constant $\gamma \triangleq \gamma_0 \,\mathrm{RMS}_i\|o_i\|_2$ for a dimensionless $\gamma_0 > 0$ (Appendix A.2), which allows us to control the tradeoff between both terms. We are left with the surrogate

$$\widehat{\mathcal{L}}(S, w, U) \triangleq \|O - P_S U\|_F^2 + \gamma^2 \,\|\mathbf{1} - P_S w\|_2^2\,, \tag{4}$$

where $P_S \triangleq P_{:,S} \in \mathbb{R}^{N\times t}$. Stacking $Y \triangleq [O \quad \gamma\mathbf{1}] \in \mathbb{R}^{N\times(d_v+1)}$ and the corresponding unknowns as $B \triangleq [U \quad \gamma w] \in \mathbb{R}^{t\times(d_v+1)}$, and adding a small ridge penalty, gives a single ridge regression,

$$\widehat{\mathcal{L}}_\lambda(S, B) \triangleq \|Y - P_S B\|_F^2 + \lambda\|B\|_F^2, \qquad \widehat{\mathcal{L}}_\lambda^\star(S) \triangleq \min_B \widehat{\mathcal{L}}_\lambda(S, B), \tag{5}$$

in which each column of B fits the corresponding column of Y and the columns interact only through the shared P_S. They are therefore all solved at once by

$$B_S \triangleq A_S^{-1} P_S^\top Y, \qquad \text{with} \quad A_S \triangleq P_S^\top P_S + \lambda I, \tag{6}$$

so one $t\times t$ inverse serves all d_v+1 targets, whatever the value dimension. We will see in Section 3.3 that this is crucial for the efficiency of KV-SURGEON, as it allows us to reuse the same inverse for every deletion. Here the ridge term λ ensures that A_S is invertible.

Note that we drop the constraint $w > 0$ in order to make (5) unconstrained and hence solvable in closed form. For KV-SURGEON we later ensure positivity, see Section 3.4. In practice, fewer than half a percent of retained entries end with a negative weight in the first place (Appendix H).

3.3 GREEDY ELIMINATION WITH CLOSED-FORM COMPENSATION

We can score a support S by $\widehat{\mathcal{L}}_\lambda^\star(S)$, i.e., the surrogate after the retained entries have been optimally refit. Minimizing it over all supports of size t remains a combinatorial problem. In order to approximate the solution, KV-SURGEON proceeds greedily: it iteratively deletes one entry whose removal increases $\widehat{\mathcal{L}}_\lambda^\star$ the least.

Deleting entry $j \in S$ amounts to constraining row j of B_S to zero, since a zero row makes column j of P_S irrelevant and contributes nothing to the ridge penalty. The cost s_j of removing entry j and the corresponding optimal update to B_S are

$$s_j \triangleq \widehat{\mathcal{L}}_\lambda^\star(S\backslash\{j\}) - \widehat{\mathcal{L}}_\lambda^\star(S) = \frac{\|B_{S,j}\|_2^2}{[A_S^{-1}]_{jj}}, \qquad \Delta B_S = -\frac{A_S^{-1} e_j B_{S,j}}{[A_S^{-1}]_{jj}}, \tag{7}$$

where $B_{S,j}$ is row j of B_S and e_j is the j-th standard basis vector. The derivation is given in Appendix A.3. The update ΔB_S is the adjustment of the surviving entries that best absorbs the deletion, and the saliency s_j is the increase in loss that this adjustment cannot remove, so compensation is already priced into the score by which entries are ranked. Applying $B_S \leftarrow B_S + \Delta B_S$ and dropping row j leaves exactly the ridge solution (6) on $S\backslash\{j\}$, and A_S^{-1} on that support follows by a rank-one downdate, so the Gram matrix never has to be re-formed or re-inverted and the saliencies of the next deletion are computed for compensated weights and values.

3.4 FINAL STEPS

Once the support S is fixed we refit the retained weights w, as in Zweiger et al. (2026). We first solve for the weights under a box constraint,

$$w \leftarrow \operatorname*{arg\,min}_{e^{-\beta} \le w \le e^{\beta}} \|\mathbf{1} - P_S w\|_2^2\,, \tag{8}$$

where $\beta > 0$ is a hyperparameter, set to $\beta = 3$ throughout (Appendix B). We write the constraint as a box on w because the quantity that is served is $\log w$: the compacted cache is run by adding $\log w_j$ to the logit of entry j, so $e^{-\beta} \le w_j \le e^{\beta}$ is the symmetric bound $|\log w_j| \le \beta$ on how far compaction may bias any single entry. The box restores the positivity relaxed in Section 3.2 and

Algorithm 1 KV-SURGEON for one KV head

Require: keys $K \in \mathbb{R}^{T \times d_k}$, values $V \in \mathbb{R}^{T \times d_v}$, calibration queries $Q \in \mathbb{R}^{N \times d_k}$, budget t, pool factor c, block size n_b, ridge λ, mass-term scale γ_0, weight bound β

$P \leftarrow \mathrm{softmax}\big(QK^\top/\sqrt{d_k}\big), \quad O \leftarrow PV$
$s_j^{\mathrm{LOO}} \leftarrow \sum_i \frac{P_{ij}^2}{(1-P_{ij})^2}\,\|v_j - o_i\|_2^2 \quad$ for $j = 1, \dots, T$
$S \leftarrow$ indices of the ct largest s_j^{LOO} ▷ Fix candidate pool
$\gamma \leftarrow \gamma_0\,\mathrm{RMS}_i\|o_i\|_2, \quad Y \leftarrow [\,O \;\; \gamma\mathbf{1}\,]$
$A^{-1} \leftarrow (P_S^\top P_S + \lambda I)^{-1}$
$B \leftarrow A^{-1} P_S^\top Y$
while $|S| > t$ **do**
  $s_j \leftarrow \|B_j\|_2^2 \,/\, [A^{-1}]_{jj} \quad \forall j \in S$
  $J \leftarrow \arg\min_{j \in S}^{n_b} s_j$ ▷ the n_b smallest saliencies
  $B \leftarrow B - A^{-1}_{:,J}\left([A^{-1}]_{JJ}\right)^{-1} B_J$
  $A^{-1} \leftarrow A^{-1} - A^{-1}_{:,J}\left([A^{-1}]_{JJ}\right)^{-1} A^{-1}_{J,:}$
  $S \leftarrow S \setminus J$ ▷ drop J from B and A^{-1}
end while
$w \leftarrow \arg\min_{e^{-\beta} \le w \le e^{\beta}} \|P_S w - \mathbf{1}\|_2^2$
$\widehat{P} \leftarrow \mathrm{diag}(P_S w)^{-1} P_S\,\mathrm{diag}(w)$
$\widehat{V} \leftarrow \arg\min_{\widehat{V}} \|O - \widehat{P}\,\widehat{V}\|_F^2$
return S, w, $\widehat{V}$

keeps the bias on the scale of the logits, so that no retained entry can be driven to dominate the softmax or to vanish from it, while the objective itself drives the retained mass to $\mu_i \approx 1$. Then holding w fixed, we reconstruct $\widehat{V}$ as

$$\widehat{V} \leftarrow \arg\min_{\widehat{V}} \left\| \widehat{P}\widehat{V} - O \right\|_F^2, \qquad \widehat{P}_{ij} \triangleq w_j P_{ij}/\mu_i, \tag{9}$$

where $\mu_i = P_{i,S}w$ is evaluated at the refit weights, so that (9) minimizes the exact error (2) for that S and w. Reweighting changes the attention pattern the stored values have to match, so reconstruction $\widehat{V}$ has to come second (Appendix G.1). The compacted cache is then served as the retained keys $\widehat{K}$ with $\log w$ added to the logits and the values $\widehat{V}$.

3.5 ALGORITHM DETAILS

We describe three important implementation details that make the method efficient in practice.

Candidate pool. Running this elimination from all T entries is unnecessarily expensive, so we first prune to a pool of $m = \lceil ct \rceil$ candidates with a pool factor $c > 1$, using the exact single-deletion score

$$s_j^{\mathrm{LOO}} \triangleq \sum_{i=1}^{N} \frac{P_{ij}^2}{(1-P_{ij})^2}\,\|v_j - o_i\|_2^2, \tag{10}$$

derived in Appendix A.4, which generalizes the eviction score of Goel et al. (2026) and An et al. (2026) to the offline setting. We keep the m entries with the largest scores. This drastically reduces the runtime of the method at marginal cost: eliminating from the full cache instead does not improve accuracy and multiplies compaction time by 3.8 (Appendix D.3). We also use (10) as the score of our ablation method LOO (cf. Section 4.3) and of the fallback below.

Block deletions. Instead of deleting a single entry per iteration, we delete a block J of the n_b entries with the smallest saliencies, written $\arg\min^{n_b}$ in Algorithm 1. This replaces n_b rank-one updates by one rank-n_b update and divides the number of steps by n_b, improving the runtime of the method. The result is again exactly the solution (6) on $S \setminus J$, as it would be after n_b successive single deletions. The only approximation is that the n_b saliencies are compared before any entry of the block has been removed (Appendix A.3).

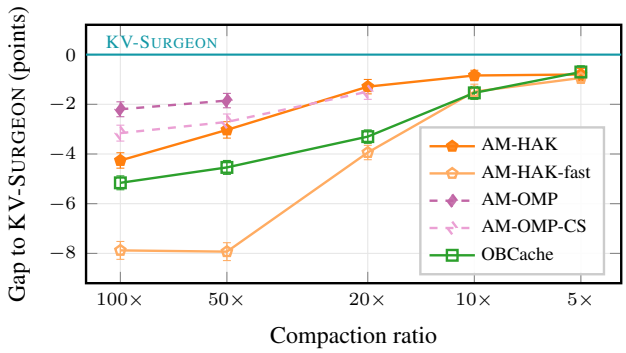


Figure 2: Mean accuracy over the eight pairs of Table 1 relative to KV-SURGEON. Error bars are one standard error of the difference. AM-OMP and AM-OMP-CS are shown only at the ratios their solver reaches on every model and benchmark.

Plain-eviction fallback. Heads whose budget exceeds a fraction τ of their cache, $t > \tau T$ with $\tau = 0.3$, are served by plain eviction on (10): eviction is near-lossless at such mild compaction, while refitting adds 27% to pipeline time (Appendix D.3). The $O(t^3)$ solves dominate pipeline cost and are largest exactly for these heads, and as t approaches the number of calibration queries N the value solve of (9) becomes underdetermined.

4 EXPERIMENTS

We compare KV-SURGEON to existing methods and show that it improves the cost–accuracy frontier across different models, benchmarks, and compaction ratios. We then analyze which part of the algorithm is the source of this improvement through controlled ablations. Finally, we study how much of the gain is retained when self-study calibration is replaced by a single repeat pass. Our code will be publicly released to ensure reproducibility.

4.1 EXPERIMENTAL SETUP

Benchmarks and models. We evaluate offline KV-cache compaction on long-context multiple-choice question-answering tasks from QuALITY (Pang et al., 2022) and LongHealth (Adams et al., 2024), and code completion tasks from Long Code Arena (Bogomolov et al., 2024), where each compacted context is reused across many questions. We evaluate five models, Qwen3-4B, Qwen3-4B-Instruct-2507, Llama-3.1-8B-Instruct, Gemma-3-12B, and Qwen3-30B-A3B (Yang et al., 2025; Grattafiori et al., 2024; Team et al., 2024), spanning three model families, dense and mixture-of-experts architectures, and different attention variants. We consider $\{100\times, 50\times, 20\times, 10\times, 5\times\}$ compaction, with $50\times$ as the primary operating point. Dataset construction and evaluation details are given in Appendix B.

Protocol. All accuracies are means over five random seeds, with one standard error reported in the tables. All methods shown in the main comparison, except LLM summarization, use heterogeneous allocation of the global cache budget across heads. The allocation is computed once per model on data disjoint from the evaluation benchmarks following Zweiger et al. (2026) and kept fixed between methods. All methods using calibration additionally share the bias parameterization and final cache refit of Section 3, so their comparisons isolate differences in support selection. KV-SURGEON hyperparameters are fixed once on Qwen3-4B with QuALITY at $50\times$ compaction and reused for all other models, benchmarks, and compaction ratios, i.e., KV-SURGEON does not require per-model or per-benchmark tuning.

Calibration and cost. To collect calibration queries, we use two different approaches. Cheap calibration queries are obtained by a *repeat pass*, which concatenates the context with a "repeat verbatim" prompt and another copy of the context (Kim et al., 2026). For *self-study*, the more expensive approach, we prompt the model to discuss the context multiple times and record the queries produced during answer generation (Eyuboglu et al., 2026). The suffix *-fast* denotes repeat-pass

Table 1: Accuracy (%) at 50× compaction. Entries are means over five seeds with standard error over seeds in grey. Within each group and column, bold denotes the highest mean and underlining denotes results within two standard errors of the highest mean. The *Mean* column contains the means over the eight model–benchmark pairs, and *Time* is the geometric mean of pipeline time per context in seconds.

Method	Time (s)	Qwen3-4B		Llama-3.1-8B		Gemma-3-12B		Qwen3-30B		Mean
		QuAL.	LongH.	QuAL.	LongH.	QuAL.	LongH.	QuAL.	LongH.	
Repeat-pass										
KV-SURGEON-fast (ours)	54	**65.9** ±0.4	**66.9** ±0.5	**61.0** ±0.4	**61.5** ±0.6	**68.1** ±0.5	**52.1** ±0.4	**71.4** ±0.3	**64.1** ±0.2	**63.9** ±0.15
OBCache (adapted)	58	62.4 ±0.7	62.8 ±0.9	59.8 ±0.3	52.4 ±0.2	66.6 ±0.4	50.2 ±0.8	67.3 ±0.4	62.8 ±0.4	60.5 ±0.20
AM-HAK-fast	54	58.1 ±0.5	53.5 ±1.3	58.7 ±0.6	53.5 ±1.1	62.2 ±0.4	45.6 ±1.2	65.8 ±0.5	59.8 ±0.6	57.1 ±0.30
Self-study										
KV-SURGEON (ours)	145	**66.8** ±0.3	**67.8** ±0.8	**62.3** ±0.5	**62.1** ±0.3	67.5 ±0.7	**56.4** ±0.7	**72.5** ±0.6	**65.2** ±0.4	**65.1** ±0.20
LOO	139	65.6 ±0.2	64.2 ±1.2	61.2 ±0.7	59.1 ±0.6	65.1 ±0.5	49.6 ±0.7	69.6 ±0.3	62.3 ±0.7	62.1 ±0.24
AM-HAK	138	64.1 ±0.9	65.0 ±0.8	60.4 ±0.5	58.2 ±0.5	65.1 ±0.3	52.7 ±1.4	69.5 ±0.3	61.4 ±0.4	62.0 ±0.26
AM-OMP-CS	266	65.7 ±0.4	65.9 ±1.1	60.6 ±0.8	55.8 ±0.9	67.2 ±0.2	50.9 ±0.8	69.7 ±0.5	63.1 ±0.2	62.4 ±0.24
AM-OMP	763	66.4 ±0.2	65.5 ±0.6	61.3 ±0.6	57.3 ±0.9	**67.7** ±0.3	54.5 ±0.9	69.2 ±0.4	63.9 ±0.6	63.2 ±0.22
Q-Filters (query-free)	1.6	52.4 ±0.5	43.6 ±0.4	47.7 ±0.8	36.1 ±1.1	55.9 ±0.3	39.0 ±0.7	50.9 ±0.3	43.2 ±0.5	46.1 ±0.22
Summarization	5.1	51.2 ±0.5	28.0 ±0.7	46.7 ±0.3	31.5 ±0.7	50.2 ±0.4	39.4 ±0.4	54.2 ±0.6	47.2 ±0.6	43.6 ±0.19
Full context	–	72.1 ±0.3	80.5 ±0.8	62.7 ±0.5	73.0 ±0.8	70.6 ±0.4	74.5 ±0.4	80.7 ±0.4	78.3 ±0.5	74.0 ±0.19

calibration for KV-SURGEON and AM-HAK. AM-OMP-CS is the cheap-solver setting of Zweiger et al. (2026), named AM-OMP-fast in the original work, and uses self-study queries. Pipeline time covers calibration-query generation and compaction, but excludes the context prefill shared by all methods and preprocessing performed once per model. Further timing and calibration details are given in Appendix B.

Baselines. Our main prior baseline is attention matching (Zweiger et al., 2026), for which we evaluate both AM-HighestAttentionKeys (AM-HAK) and AM-OMP using the authors' implementation and hyperparameters. We also adapt several serving-time eviction methods to the offline setting (Appendix F) and include our adaptation of OBCache in the main experiments. LOO ranks entries by the exact single-deletion score of (10) and shares KV-SURGEON's calibration queries, head-budget allocation, and final refit, providing a controlled comparison to independent selection. We additionally include LLM summarization and query-free baselines (Appendix C).

4.2 KV-SURGEON IMPROVES THE COST–ACCURACY FRONTIER

At 50× compaction, both KV-SURGEON and KV-SURGEON-fast exceed the prior state of the art (Table 1). KV-SURGEON-fast reaches 63.9% mean accuracy in 54 seconds per context, compared with 63.2% for AM-OMP in 763 seconds, a 14.1× reduction in pipeline time. Self-study calibration improves KV-SURGEON further to 65.1% in 145 seconds, 1.9 percentage points above AM-OMP at 5.3× lower time. At comparable cost, KV-SURGEON-fast is 3.3 percentage points ahead of adapted OBCache, while KV-SURGEON is 3.0 points ahead of AM-HAK.

These gains are consistent across models and benchmarks. For both calibration methods, KV-SURGEON is best or tied for best within uncertainty on all eight model–benchmark pairs, while KV-SURGEON-fast is ahead of every other method beyond uncertainty. Thus, the more expensive AM-OMP variants no longer lie on the cost–accuracy frontier at our primary operating point (Figure 1).

The advantage of KV-SURGEON also persists across compaction ratios. Figure 2 shows mean accuracy over the eight model–benchmark pairs relative to KV-SURGEON. All prior methods are worse by more than two standard errors at every compaction ratio. The gap is largest between 100× and 20× compaction and narrows at 10× and 5×, where all methods approach the performance of the full-context ceiling. We therefore focus the analysis below on large compaction ratios, where support selection has the largest effect.

The same compute time advantage transfers to project-level code completion on Long Code Arena (Table 12, Appendix C.1). At 50×, KV-SURGEON and KV-SURGEON-fast match AM-OMP within uncertainty while using 9.7 and 21.5 times less compute time, respectively. At 20×, KV-SURGEON

Table 2: Qwen3-4B on QuALITY (first 25 articles) at $50\times$ compaction. We compare the impact of compensation during selection. We vary the selection rule, i.e., the saliency by which entries are ranked for deletion, and whether the surviving entries are updated after each deletion.

Method	Selection rule	Compensation	Accuracy (%)
KV-SURGEON	Equation (7)	✓	**66.9**
KV-SURGEON w/o compensation	Equation (7)	✗	63.9
LOO	Equation (10)	✗	64.7

has the highest point estimate, with AM-OMP-CS the only prior method within uncertainty at 6.7 times higher cost.

4.3 SOURCES OF THE PERFORMANCE GAIN

The gain over independent selection comes from compensating the surviving cache entries as the support changes. To isolate this effect, we keep the surrogate, candidate pool, calibration queries, and final refit fixed and disable only the compensating update: we delete entries based on (7) but do not apply the ΔB_S update to the remaining entries. At $50\times$ compaction on Qwen3-4B with QuALITY, this reduces accuracy from 66.9% to 63.9% (Table 2). KV-SURGEON without compensation also performs similarly to LOO, which uses a different deletion rule ((10)) and does not adapt the remaining entries at all. Replacing the saliency score alone does not explain the improvement. Rather, the gain comes from updating the remaining entries *during* selection. Additional controls that remove the opportunity for iterative compensation lead to the same conclusion (Appendices D and E).

The improvement is also not an artifact of the optimized allocation of the global cache budget across heads inherited from Zweiger et al. (2026). Under uniform allocation, KV-SURGEON remains 3.0 accuracy points ahead of AM-HAK (Appendix D). Compensation-aware support selection therefore provides an improvement that is orthogonal to that of the offline-compaction pipeline proposed by Zweiger et al. (2026).

4.4 CHEAP CALIBRATION PRESERVES MOST OF THE GAIN

KV-SURGEON is comparatively insensitive to how calibration queries are collected. Replacing self-study with a single repeat pass reduces pipeline time from 145 to 54 seconds per context at $50\times$ compaction while reducing mean accuracy by only 1.2 points. The same change is substantially more costly for AM-HAK, which loses 4.9 points (Table 1). This makes KV-SURGEON-fast not only cheaper, but sufficient to outperform all prior methods at the primary operating point.

This robustness persists across compaction ratios. Replacing self-study with repeat-pass calibration costs KV-SURGEON at most 1.2 points of mean accuracy at any compaction ratio, with the gap shrinking as more cache entries are retained. Detailed results across ratios are reported in Appendix C; Appendix E offers a hypothesis for this robustness.

5 CONCLUSION

Prior offline KV-cache compaction methods first deleted cache entries and then adapted the remaining ones. We showed that coupling the deletion of entries and the adaptation of the remaining ones by performing those steps jointy can be performed efficiently and leads to improved performance. We derived a quadratic surrogate of the attention-output error that allows for an efficient greedy algorithm with compensation-aware selection: every candidate deletion is scored after the retained entries have compensated for it. The resulting method, KV-SURGEON, improves the cost–accuracy frontier at every compaction ratio on QuALITY, LongHealth, and Long Code Arena with models from 4B to 30B parameters, and exceeds all prior methods at two fifths of the cost of the strongest baseline. A controlled comparison attributes the gain over independent selection to compensation during selection, not to the surrogate.

USE OF GENERATIVE AI

In this work, we made use of AI tools to produce and refactor code for the experiments and generate figures, search for related work, and help with writing the article. We have not used LLMs for research ideation or for the execution of the experiments. The method, the mathematical results, and the experimental protocol were developed by the authors. We have reviewed all AI-assisted work. We take full responsibility for the final content of this work.

REFERENCES

Lisa Adams, Felix Busch, Tianyu Han, Jean-Baptiste Excoffier, Matthieu Ortala, Alexander Löser, Hugo JWL. Aerts, Jakob Nikolas Kather, Daniel Truhn, and Keno Bressem. Longhealth: A question answering benchmark with long clinical documents, 2024. URL https://arxiv.org/abs/2401.14490.

Yongqi An, Chang Lu, Kuan Zhu, Tao Yu, Chaoyang Zhao, Hong Wu, Ming Tang, and Jinqiao Wang. Rest-kv: Robust kv cache eviction with layer-wise output reconstruction and spatial-temporal smoothing. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 86943–86967, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/8be9c134bb193d8bd3827d4df8488228-Paper-Conference.pdf.

Egor Bogomolov, Aleksandra Eliseeva, Timur Galimzyanov, Evgeniy Glukhov, Anton Shapkin, Maria Tigina, Yaroslav Golubev, Alexander Kovrigin, Arie Van Deursen, Maliheh Izadi, et al. Long code arena: a set of benchmarks for long-context code models. *arXiv preprint arXiv:2406.11612*, 2024. URL https://arxiv.org/pdf/2406.11612.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed Abdelfattah, and Kai-Chiang Wu. Palu: Kv-cache compression with low-rank projection. In *International Conference on Learning Representations*, volume 2025, pp. 50222–50249, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/7da6e0e00702c60607a6ae05c802ef85-Paper-Conference.pdf.

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 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18476–18499, 2024. URL https://aclanthology.org/2024.emnlp-main.1027.pdf.

Sabri Eyuboglu, Ryan Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Liu, Atri Rudra, James Y Zou, Azalia Mirhoseini, and Christopher Re. Cartridges: Lightweight and general-purpose long context representations via self-study. In *International Conference on Learning Representations*, volume 2026, pp. 42642–42687, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/4681359a7b1e94571598ad1adda35e6e-Paper-Conference.pdf.

Yuan Feng, Junlin Lv, Haoyu Guo, Yukun Cao, S Kevin Zhou, and Xike Xie. Criticalkv: Optimizing kv cache eviction from an output perturbation perspective, 2026. URL https://openreview.net/pdf?id=vMwlcLyI4b.

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. URL https://arxiv.org/pdf/2503.02812.

Raghavv Goel, Junyoung Park, Mukul Gagrani, Dalton Jones, Matthew J Morse, Matthew Harper Langston, Christopher Lott, and Mingu Lee. Caote: Optimizing kv cache memory through attention output error-based token eviction. In *ICLR 2026 Workshop on Memory for LLM-Based Agentic Systems*, 2026. URL https://openreview.net/pdf?id=yolbP0NoZv.

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, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jen-

nifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models. July 2024. URL https://arxiv.org/abs/2407.21783.

Yuzhe Gu, Xiyu Liang, Jiaojiao Zhao, and Enmao Diao. Obcache: Optimal brain kv cache pruning for efficient long-context llm inference. 2026. URL https://openreview.net/pdf?id=f58VZpG6ls.

Zhiyu Guo, Hidetaka Kamigaito, and Taro Watanabe. Attention score is not all you need for token importance indicator in kv cache reduction: Value also matters. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21158–21166, 2024. URL https://aclanthology.org/2024.emnlp-main.1178.pdf.

Babak Hassibi and David Stork. Second order derivatives for network pruning: Optimal brain surgeon. In S. Hanson, J. Cowan, and C. Giles (eds.), *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann, 1993. URL https://proceedings.neurips.cc/paper/1992/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun S Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *Advances in Neural Information Processing Systems*, 37:1270–1303, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/028fcbcf85435d39a40c4d61b42c99a4-Paper-Conference.pdf.

Jang-Hyun Kim, Jinuk Kim, Sangwoo Kwon, Jae W Lee, Sangdoo Yun, and Hyun Oh Song. Kvzip: Query-agnostic kv cache compression with context reconstruction. *Advances in Neural Information Processing Systems*, 38:167563–167591, 2026. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f4eaa4b8f2d08edb3f0af990d56134ea-Paper-Conference.pdf.

Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In David S. Touretzky (ed.), *Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]*, pp. 598–605. Morgan Kaufmann, 1989. URL https://proceedings.neurips.cc/paper/1989/hash/6c9882bbac1c7093bd25041881277658-Abstract.html.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/28ab418242603e0f7323e54185d19bde-Paper-Conference.pdf.

Manlai Liang, JiaMing Zhang, Xiong Li, and Jinlong Li. Lagkv: Lag-relative information of the kv cache tells which tokens are important. *arXiv preprint arXiv:2504.04704*, 2025. URL https://arxiv.org/pdf/2504.04704.

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. *Advances in Neural Information Processing Systems*, 36:52342–52364, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a452a7c6c463e4ae8fbdc614c6e983e6-Paper-Conference.pdf.

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 Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32332–32344. PMLR, 21–27 Jul 2024. URL https://raw.githubusercontent.com/mlresearch/v235/main/assets/liu24bz/liu24bz.pdf.

Piotr Nawrot, Adrian Łańcucki, Marcin Chochowski, David Tarjan, and Edoardo Ponti. Dynamic memory compression: Retrofitting LLMs for accelerated inference. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37396–37412. PMLR, 21–27 Jul 2024. URL https://raw.githubusercontent.com/mlresearch/v235/main/assets/nawrot24a/nawrot24a.pdf.

Matanel Oren, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz. Transformers are multi-state rnns. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18724–18741, 2024. URL https://aclanthology.org/2024.emnlp-main.1043.pdf.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel R. Bowman. Quality: Question answering with long input texts, yes!, 2022. URL https://arxiv.org/abs/2112.08608.

Utkarsh Saxena, Gobinda Saha, Sakshi Choudhary, and Kaushik Roy. Eigen attention: Attention in low-rank space for kv cache compression. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15332–15344, 2024. URL https://aclanthology.org/2024.findings-emnlp.899.pdf.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. URL https://arxiv.org/pdf/2408.00118.

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

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

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pp. 21875–21895, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/5e5fd18f863cbe6d8ae392a93fd271c9-Paper-Conference.pdf.

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

Yuxin Zhang, Yuxuan Du, Gen Luo, Yunshan Zhong, Zhenyu Zhang, Shiwei Liu, and Rongrong Ji. CaM: Cache merging for memory-efficient LLMs inference. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 58840–58850. PMLR, 21–27 Jul 2024. URL https://raw.githubusercontent.com/mlresearch/v235/main/assets/zhang24n/zhang24n.pdf.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6ceefa7b15572587b78ecfcebb2827f8-Paper-Conference.pdf.

Adam Zweiger, Xinghong Fu, Han Guo, and Yoon Kim. Fast kv compaction via attention matching. 306, 2026. URL https://openreview.net/pdf?id=t01SKTj8pP.

A Derivations

A.1 The surrogate as a second-order Taylor expansion

The first relaxation of Section 3.2 replaces the weight μ_i^{-2} in (3) by one, which gives the quadratic $\mathcal{Q}(S, w, U) \triangleq \sum_{i=1}^{N} \|P_{i,S}U - \mu_i\, o_i\|_2^2$; the second and third relaxations then act on this quadratic. We show that $\mathcal{Q}$ is the second-order Taylor expansion of $\mathcal{L}$ around the original cache. The method represents a deletion by removing an index from S, and a support S and the original cache have parameters of different sizes. To place them in a common coordinate system we use, in this proof only, a full-length weight vector $m \in \mathbb{R}^T$ with $m_j = w_j$ for $j \in S$ and $m_j = 0$ otherwise, and $\widetilde{U} \in \mathbb{R}^{T \times d_v}$ with rows $m_j \widehat{v}_j$, which vanish outside S. Then $P_{i,S}U = P_i\widetilde{U}$ and $\mu_i = P_i m$, where P_i is row i of P, so that

$$\mathcal{L} = \sum_{i=1}^{N} \frac{\left\|P_i\widetilde{U} - (P_i m)\, o_i\right\|_2^2}{(P_i m)^2}, \qquad \mathcal{Q} = \sum_{i=1}^{N} \left\|P_i\widetilde{U} - (P_i m)\, o_i\right\|_2^2$$

are functions of $(\widetilde{U}, m) \in \mathbb{R}^{T \times d_v} \times \mathbb{R}^T$ whose domain contains the original cache $(\widetilde{U}, m) = (V, \mathbf{1})$. There the vector inside the norms is $P_iV - (P_i\mathbf{1})\, o_i = o_i - o_i = 0$ for every query, because $P_iV = o_i$ by the definition of O and $P_i\mathbf{1} = 1$ since every row of P is a softmax; hence $\mathcal{L}$ and $\mathcal{Q}$ vanish together with their gradients. Write $(\widetilde{U}, m) = (V + \Delta_U, \mathbf{1} + \Delta_m)$. Substituting, the vector inside the norms becomes $P_i(V + \Delta_U) - \big(P_i(\mathbf{1} + \Delta_m)\big)\, o_i = (P_iV - o_i) + P_i\Delta_U - (P_i\Delta_m)\, o_i = P_i\Delta_U - (P_i\Delta_m)\, o_i$, again by $P_iV = o_i$ and $P_i\mathbf{1} = 1$. This is linear in (Δ_U, Δ_m), so its squared norm is a homogeneous quadratic in the perturbation, while $(P_i m)^{-2} = 1 - 2P_i\Delta_m + O(\|\Delta_m\|_2^2)$. Hence $\mathcal{L} = \mathcal{Q} + O(\|(\Delta_U, \Delta_m)\|^3)$, and $\mathcal{Q}$, itself a quadratic, is the second-order Taylor polynomial of $\mathcal{L}$ at the original cache. As in OBS (Hassibi & Stork, 1993), the expansion is then used at deletions, which move m_j from one to zero and row j of $\widetilde{U}$ from v_j to zero and are not small perturbations.

For unchanged values, $\widetilde{U} = \operatorname{diag}(m)V$, the two terms inside the norms are $P_i \operatorname{diag}(m)V = \sum_j m_j P_{ij} v_j$ and $(P_i m)\, o_i = \sum_j m_j P_{ij}\, o_i$, so the vector inside the norms is $\sum_j m_j P_{ij}(v_j - o_i)$ and the surrogate is the quadratic form

$$\mathcal{Q} = m^\top H m, \qquad H_{jk} \triangleq \sum_{i=1}^{N} P_{ij}P_{ik}\,(v_j - o_i)^\top (v_k - o_i).$$

Its off-diagonal entries record whether the errors of deleting j and k reinforce or cancel. Its diagonal $H_{jj} = \sum_i P_{ij}^2 \|v_j - o_i\|_2^2$ is the second-order truncation of the exact single-deletion loss, the score of (10) without the factor $(1 - P_{ij})^{-2}$ (Appendix A.4). Finally, $H\mathbf{1} = 0$ because $\sum_k P_{ik}(v_k - o_i) = 0$. This is the null direction of the Hessian along uniform scalings of all weights, which leave the ratio of (1) unchanged; the surrogate alone therefore does not determine the scale of the weights, and the mass term of (4) does (Appendices A.2 and G.1).

A.2 Scale of the mass term

In (3) the mass error of query i enters as $(1 - \mu_i)\, o_i$, so its natural weight is $\|o_i\|_2^2$: a query with a large output is hurt more by a given relative mass deficit. Squaring the sum, as (3) does, keeps this per-query weight but also produces the cross-term $2(1 - \mu_i)\, o_i^\top (P_{i,S}U - o_i)$, which couples U and w through the query-specific vector o_i. With that term, the joint problem in (U, w) has a Hessian of size $|S|(d_v + 1)$ instead of $d_v + 1$ targets sharing one $|S| \times |S|$ Gram matrix, and the closed forms of Section 3.3 are lost. Replacing $\|o_i\|_2$ by one common scalar removes the coupling, and choosing that scalar as $\mathrm{RMS}_i\|o_i\|_2 = (N^{-1}\sum_i \|o_i\|_2^2)^{1/2}$ keeps the mass column of Y on the scale of the output columns: at $\gamma_0 = 1$, a uniform relative mass deficit $1 - \mu_i \equiv \epsilon$ costs $\gamma^2\epsilon^2 N = \epsilon^2 \sum_i \|o_i\|_2^2$, which is what the mass-error term of (3) would cost. The factor γ_0 is therefore dimensionless and of order one, and it sets the weight of the mass term against the output term.

A.3 Derivation of the compensated deletion step

We show that (7) gives the exact increase of $\widehat{\mathcal{L}}_\lambda^\star$ when entry j is removed from the support, together with the minimizer on the reduced support. The argument has three steps: the ridge objective of (5)

is an exact quadratic around its minimizer, deleting an entry is a linear constraint on the coefficient matrix, and a quadratic under one linear constraint is minimized by a single Lagrange multiplier.

Write $A \triangleq A_S$, $G \triangleq A_S^{-1}$, $X \triangleq P_S$, and $B^\star \triangleq B_S$; $|S|$ denotes the size of the current support, which shrinks from m to t during elimination, so A and G are $|S| \times |S|$. The objective of (5) is $\widehat{\mathcal{L}}_\lambda(S, B) = \|Y - XB\|_F^2 + \lambda\|B\|_F^2$. Expanding both squares at $B^\star$ gives, for every $\Delta \in \mathbb{R}^{|S|\times(d_v+1)}$,

$$\begin{aligned}\widehat{\mathcal{L}}_\lambda(S, B^\star + \Delta) =& \|Y - XB^\star\|_F^2 - 2\,\mathrm{tr}\big(\Delta^\top X^\top (Y - XB^\star)\big) + \mathrm{tr}\big(\Delta^\top X^\top X \Delta\big) \\ &+ \lambda\|B^\star\|_F^2 + 2\lambda\,\mathrm{tr}\big(\Delta^\top B^\star\big) + \lambda\,\mathrm{tr}\big(\Delta^\top \Delta\big) \\ =& \widehat{\mathcal{L}}_\lambda(S, B^\star) - 2\,\mathrm{tr}\big(\Delta^\top (X^\top Y - AB^\star)\big) + \mathrm{tr}\big(\Delta^\top A \Delta\big),\end{aligned}$$

where the second line collects the terms of the first line by their degree in Δ and uses $A = X^\top X + \lambda I$. The minimizer $B^\star = A^{-1}X^\top Y$ satisfies $AB^\star = X^\top Y$, so the term linear in Δ vanishes and

$$\widehat{\mathcal{L}}_\lambda(S, B^\star + \Delta) = \widehat{\mathcal{L}}^\star_\lambda(S) + \mathrm{tr}\big(\Delta^\top A \Delta\big)$$

holds for every Δ, not only for small ones: $\widehat{\mathcal{L}}_\lambda$ is a polynomial of degree two in B, so its expansion around any point stops after the quadratic term, and at the minimizer the linear term is zero. A matrix B with a zero row j has $\widehat{\mathcal{L}}_\lambda(S, B) = \widehat{\mathcal{L}}_\lambda(S \setminus \{j\}, B_{-j})$, where B_{-j} drops that row, since the row multiplies column j of X and contributes nothing to the ridge penalty; conversely every coefficient matrix on $S \setminus \{j\}$ arises this way. Passing to the support $S \setminus \{j\}$ is therefore the same as requiring row j of $B^\star + \Delta$ to vanish, that is, $e_j^\top \Delta = -B^\star_{j,:}$, and

$$\widehat{\mathcal{L}}^\star_\lambda(S \setminus \{j\}) - \widehat{\mathcal{L}}^\star_\lambda(S) = \min_{\Delta}\ \mathrm{tr}\big(\Delta^\top A \Delta\big) \quad \text{subject to} \quad e_j^\top \Delta = -B^\star_{j,:}.$$

With a multiplier $\nu \in \mathbb{R}^{d_v+1}$, stationarity of the Lagrangian $\mathrm{tr}(\Delta^\top A\Delta) + (e_j^\top\Delta + B^\star_{j,:})\nu$ gives $2A\Delta + e_j\nu^\top = 0$, so $\Delta = -\frac{1}{2}Ge_j\nu^\top$. Substituting into the constraint yields $\frac{1}{2}G_{jj}\nu^\top = B^\star_{j,:}$, and therefore

$$\Delta B = -\frac{Ge_jB^\star_{j,:}}{G_{jj}}, \qquad \mathrm{tr}\big(\Delta B^\top A\, \Delta B\big) = \frac{\|B^\star_{j,:}\|_2^2\; e_j^\top GAGe_j}{G_{jj}^2} = \frac{\|B^\star_{j,:}\|_2^2}{G_{jj}},$$

which is (7) with $B^\star_{j,:} = B_{S,j}$ and $G_{jj} = [A_S^{-1}]_{jj}$. As A is positive definite, the stationary point is the unique minimizer, so the remaining rows of $B^\star + \Delta B$ are the minimizer of (5) on $S \setminus \{j\}$. The inverse Gram matrix of the reduced support follows from the block-inverse identity: the inverse of A with row and column j removed equals $G - Ge_je_j^\top G/G_{jj}$ with row and column j removed, which is the downdate in Algorithm 1. The same identity gives $1/G_{jj} = A_{jj} - A_{j,-j}(A_{-j,-j})^{-1}A_{-j,j}$, the Schur complement of entry j, which equals the ridge-regularized residual $\min_c\{\|X_{:,j} - X_{:,-j}c\|_2^2 + \lambda\|c\|_2^2\} + \lambda$ of regressing the attention column of entry j on the columns of the other retained entries. This is the sense in which $1/G_{jj}$ measures what the other retained entries cannot reproduce.

Deleting a block. The same argument covers the block deletions of Section 3.5. Deleting a set $J \subset S$ of n_b entries is the constraint $E_J^\top \Delta = -B^\star_{J,:}$, where $E_J \in \mathbb{R}^{|S|\times n_b}$ collects the unit vectors e_j, $j \in J$, and $B^\star_{J,:}$ the corresponding rows of $B^\star$. With a multiplier $\nu \in \mathbb{R}^{n_b\times(d_v+1)}$, stationarity gives $\Delta = -\frac{1}{2}GE_J\nu$, the constraint gives $\frac{1}{2}G_{JJ}\nu = B^\star_{J,:}$ with $G_{JJ} \triangleq E_J^\top G E_J$ the $n_b \times n_b$ principal submatrix of G, and therefore

$$\Delta B = -G_{:,J}\, G_{JJ}^{-1} B^\star_{J,:}, \qquad \widehat{\mathcal{L}}^\star_\lambda(S \setminus J) - \widehat{\mathcal{L}}^\star_\lambda(S) = \mathrm{tr}\big(B^{\star\top}_{J,:} G_{JJ}^{-1} B^\star_{J,:}\big),$$

and the block-inverse identity gives $G - G_{:,J}G_{JJ}^{-1}G_{J,:}$, restricted to $S\setminus J$, as the inverse Gram matrix of the reduced support. These are the updates in Algorithm 1. The rows of $B^\star + \Delta B$ outside J are the minimizer of (5) on $S \setminus J$, exactly as after n_b successive single deletions in any order; the block step approximates the sequential procedure only in choosing J from saliencies computed before any of its entries is removed.

Block deletions. Deleting a block $J \subseteq S$ at once (Section 3.5) is the same argument with the constraint $E_J^\top \Delta = -B^\star_{J,:}$, where E_J collects the columns of the identity indexed by J and $B^\star_{J,:}$ the

corresponding rows of $B^\star$. Stationarity of the Lagrangian gives $\Delta = -\frac{1}{2} G E_J \Lambda^\top$ for a multiplier $\Lambda \in \mathbb{R}^{(d_v+1)\times|J|}$, and the constraint fixes $\frac{1}{2} G_{JJ} \Lambda^\top = B^\star_{J,:}$, so that

$$\Delta B = -G_{:,J} \, (G_{JJ})^{-1} B^\star_{J,:}, \qquad F^\star(S \setminus J) - \widehat{\mathcal{L}}^\star_\lambda(S) = \operatorname{tr}\big(B^{\star\top}_{J,:} (G_{JJ})^{-1} B^\star_{J,:}\big),$$

using $G_{J,:} A G_{:,J} = (GAG)_{JJ} = G_{JJ}$, with G_{JJ} the principal submatrix of G on J and $G_{:,J}$ its columns indexed by J. This is the update of Algorithm 1, and the inverse on the reduced support is the block downdate $G - G_{:,J}(G_{JJ})^{-1} G_{J,:}$ with the rows and columns of J removed. Coefficients and inverse are therefore exactly those of the ridge solution on $S \setminus J$, as they would be after $|J|$ single deletions; the block step approximates the elimination only in that all entries of J are ranked by saliencies computed before any of them is removed.

Cost. Forming the Gram matrix of the pool of $m = \lceil ct \rceil$ candidates costs $O(Nm^2)$ and its inverse $O(m^3)$. Each single-entry elimination step costs $O(|S|^2 + |S| d_v)$ for the saliencies, the coefficient update, and the downdate, so the $m - t$ steps cost $O(m^3 + m^2 d_v)$ per head in total. Deleting a block J of n_b entries per iteration replaces n_b rank-one downdates by one rank-n_b downdate and one inverse of the $n_b \times n_b$ matrix G_{JJ}. The arithmetic count is the same as for n_b single deletions, but the saliencies are computed once per block rather than once per deletion, and the updates run as matrix products with n_b columns.

A.4 The single-deletion score

The score of (10) is the one case in which the exact loss of (2) has a closed form: a single deletion with every other cache parameter at its original value, $S = \{1, \dots, T\} \setminus \{j\}$, $w = \mathbf{1}$, and $\widehat{v}_k = v_k$ for $k \in S$. Then $U = V_S$, $\mu_i = 1 - P_{ij}$, and $P_{i,S} U - \mu_i o_i = (o_i - P_{ij} v_j) - (1 - P_{ij})\, o_i = -P_{ij}(v_j - o_i)$, so (2) gives

$$\widehat{o}_i - o_i = -\frac{P_{ij}}{1 - P_{ij}} \, (v_j - o_i), \qquad \mathcal{L} = \sum_{i=1}^{N} \frac{P_{ij}^2}{(1 - P_{ij})^2} \|v_j - o_i\|_2^2 = s_j^{\mathrm{LOO}}.$$

Goel et al. (2026) and An et al. (2026) derive the per-query error for serving-time eviction, with queries read from the prompt suffix; summing its squared norm over the calibration queries carries it to the offline setting. The error is the product of two factors. The renormalization factor $P_{ij}/(1 - P_{ij})$ is large for a heavy hitter (Zhang et al., 2023) with P_{ij} near one, because the mass of the deleted entry is redistributed over retained entries that hold little mass themselves. The value novelty $v_j - o_i$ is small when the entry carries what the head already retrieves, so even a heavily attended entry can be deleted at no cost if its value is close to the current output. KV-SURGEON uses the score in two roles only: as the prefilter that forms the candidate pool, and, followed by the same final refit, as the controlled baseline LOO of Section 4.3. Its second-order truncation, the diagonal H_{jj} of Appendix A.1, drops the factor $(1 - P_{ij})^{-2}$.

B Experimental details

This section records what is needed to reproduce and read the experiments; Section 4.1 gives the summary. Appendix B.1 states the protocol and conventions, Appendix B.2 breaks down where the pipeline time goes.

B.1 Protocol and conventions

Task and benchmarks. The model reads a context once, the resulting cache is compacted once by a target factor between $5\times$ and $100\times$, and all benchmark questions about that context are answered from the compacted cache, so calibration and compaction are one-time costs per context. QuALITY uses the dev split, articles 0–49, with $n = 894$ multiple-choice questions; one article is one context. LongHealth consists of 20 fictional patient records, grouped into four contexts of five patients with roughly 60k tokens each; each record is compacted as its own chunk at the same ratio and the compacted chunks are concatenated (Appendix G), and all $n = 400$ multiple-choice questions are evaluated against their contexts. Accuracy is the percentage of correctly answered questions. Answers are extracted from the generated text, and questions whose answer cannot be parsed count as incorrect; parse rates are checked per run and are uniformly high in the reported cells, so parsing does not drive the reported differences. All eight model and benchmark combinations are reported.

Compaction ratio and per-head budgets. The compaction factor, $5\times$ to $100\times$, is the nominal ratio of the original cache size to the compacted one. Cache-side methods are reported at the nominal ratio because measured effective ratios agree with it to within one percent. Heads differ widely in how much compaction they tolerate, so the compaction factor is a global budget rather than a per-head prescription. Head (ℓ, h) receives its own ratio $r_{\ell h}$, subject to the constraint that $\mathrm{mean}_{\ell h}\, r_{\ell h}$ equals the reciprocal of the compaction factor; each head is additionally capped at $r_{\ell h} \leq 0.75$. The allocation is computed once per model on contexts disjoint from the benchmark contexts, following the procedure introduced by Zweiger et al. (2026). We measure per head how the log-perplexity of model-generated reference answers degrades as that head alone is compressed, and we allocate the budget along these influence curves. The solver is ratio-agnostic: it finds one set of per-head proportions that works across all target ratios, and this single allocation is rescaled to each compaction factor. Every method in a table consumes the same allocation, so differences between rows are attributable to selection and refit alone; KVzip in its published configuration is the deliberate exception (Appendix F). Uniform per-head budgets serve as an ablation (Appendix D).

Calibration tiers. All implemented methods score positions against calibration queries recorded from the model itself and, except AM-OMP, come in two tiers that differ only in how these queries are generated. The fast tier prompts the model to reproduce the context and records the query vectors of that single pass. The self-study tier additionally generates self-study conversations about the context, which cost more time but match the future workload better. AM-HAK-fast and KV-SURGEON-fast are the fast variants of AM-HAK and KV-SURGEON. AM-OMP, AM-OMP-CS, and KV-SURGEON are self-study methods; AM-OMP-CS is the cheap-solver setting of Zweiger et al. (2026), named AM-OMP-fast there, which selects several keys per greedy step and refits less often. After selecting which positions to keep, every calibrated method applies the same support-fixed refit of the retained weights and values (Section 3.4), so accuracy differences between them reflect the selection rule. AM-OMP is evaluated only at the smallest ratios because its runtime grows steeply with the retention budget.

Replication and uncertainty. Each cell is the mean over five replicate runs with explicit seeds, and the grey number is the standard error over these runs; a single run is noisy to about ± 1.5 points at $n = 894$ and ± 2.0 at $n = 400$, comparable to many method gaps, so we only trust orderings that survive the seed error bars. These standard errors measure generation noise under fixed contexts and questions, not uncertainty over the population of contexts, and LongHealth contributes only four composite contexts; generalization rests on the spread of the margins across the eight combinations in Appendix C. Full-context ceilings come from seed-replicated runs without compaction under the same protocol.

Timing conventions. Pipeline time is the average wall-clock per context spent on calibration query generation plus compaction, in seconds. It excludes prefilling the context to extract its cache, answer generation, and answer extraction, all of which are common to all methods. Query-free methods therefore report compaction time alone. All runs are single-job SLURM runs launched with the same tooling, so times are comparable within a table; comparisons across models should be treated as indicative. Every table splits calibrated methods by calibration tier, repeat-pass above self-study; AM-OMP-CS uses self-study queries and sits in the self-study group.

Defaults. The method defaults were fixed once on Qwen3-4B with QuALITY at $50\times$ and reused unchanged for every other model, benchmark, and ratio: the mass-term scale $\gamma_0 = 1$, so that γ is the RMS norm of the full-cache outputs; the ridge $\lambda = 0.01$, applied relative to the mean diagonal of $P_S^\top P_S$ rather than as an absolute penalty; the block size $b = 32$ deletions per elimination step; the candidate pool of $4t$; the keep-ratio threshold of 0.3 above which a head is left to plain eviction; and, in the final refit, the bound $|\log w_j| \leq 3$ on the served logit biases, that is, $w_j \in [e^{-3}, e^3]$, the interval that Table 17 refers to. Appendix D varies γ_0, the pool, and the threshold.

B.2 WHERE THE TIME GOES

Query generation sets the cost of the self-study tier. Self-study query generation is a sampled generation pass over the context and costs the same whichever method consumes the queries: 29 to 33 seconds per QuALITY article and 193 to 341 per LongHealth context on the dense models,

3.4 to 5.8 ms per token, and 162 and 1062 seconds on Qwen3-30B-A3B, whose mixture-of-experts decoding runs four to five times slower per token under vLLM. A repeat pass, including re-extraction of its query vectors, costs 5 to 160 seconds at 50×, and the query-free methods need no pass at all. Table 3 splits the self-study pipeline for LOO, KV-SURGEON, and AM-HAK, which consume the same queries. At 50× compaction is 3 to 25% of the pipeline for all three, so the horizontal spread between them in Figure 1, 2 to 12%, comes entirely from the compaction step, while the order-of-magnitude gaps are between tiers.

Compaction grows with the retained fraction. The compaction step of KV-SURGEON costs 1.4× that of LOO at 50× and 1.8 to 2.3× at 5×, because the block solve grows with the candidate pool: the log-log slope of compaction time against the retained fraction is 0.45 to 0.6 for LOO and AM-HAK and 0.65 to 0.95 for KV-SURGEON, and since query generation does not depend on the compaction factor, the compaction share rises from about 20% to 50–60% across the grid on the dense models. Gemma-3-12B's compaction is small because five of every six layers use sliding-window attention and only the global layers are compacted.

	Query gen.	LOO		KV-SURGEON		AM-HAK	
Model, benchmark	(s)	50×	5×	50×	5×	50×	5×
Qwen3-4B, QuALITY	33	7.7	29	10.5	53	7.0	28
Llama-3.1-8B, LongHealth	274	66	284	93	578	57	277
Gemma-3-12B, QuALITY	29	1.2	4.8	1.7	11	1.1	4.8
Qwen3-30B-A3B, QuALITY	162	5.5	20	–	–	4.6	20
Qwen3-30B-A3B, LongHealth	1062	47	–	65	–	–	–

Table 3: Where self-study pipeline time goes. Query generation is the sampled self-study pass, in seconds per context; it does not depend on the method or on the compaction factor. The remaining columns are the compaction step alone, in seconds per context, at 50× and 5× for the three methods that consume the same queries. Means over all seeded runs; '–' marks cells not measured.

C FULL RESULTS

The tables that follow cover one model and benchmark combination each and report QA accuracy and pipeline time for every method at all five compaction ratios, under the protocol of Appendix B. The conventions are the same throughout and are stated once here. Rows are grouped as in Table 1, query-free, repeat-pass, and self-study, separated by midrules, and the best accuracy per ratio within each group is bold; an entry is underlined when it trails the bold entry by at most two standard errors of the difference, computed from the two seed standard errors, so that bold without any underlined entry in the same group and column marks a separation beyond seed noise. Time is pipeline time per context in seconds; LongHealth contexts are about eight times longer than QuALITY articles, so every method costs roughly an order of magnitude more there. '–' marks ratios a method does not reach: AM-OMP and AM-OMP-CS are capped because their solver cost grows with the retention budget. Summarization has no size parameter and therefore no table row.

Observations across the grids. Three regularities are not visible in Table 1. First, mild compaction saturates: averaged over the eight combinations, the self-study methods span 54.7 to 59.3 points at 100× but only 72.9 to 73.7 at 5× against a ceiling of 74.0, and Llama-3.1-8B on QuALITY reaches its ceiling of 62.7% at 20×, so that combination is informative only at 50× and 100×. Second, KV-SURGEON is never behind at the cell level: it is bold in 32 of the 40 self-study cells and within noise of the best in the other eight, and its largest deficit anywhere is 1.3 points, to an AM-OMP variant costing far more. Third, KV-SURGEON-fast is bold or within noise in 37 of the 40 repeat-pass cells and separated from every other repeat-pass method in all 16 cells at 50× and 100×; its three losses beyond noise are all at 10× and 5× and at most 1.8 points.

Query-free baselines. Q-Filters, StreamingLLM, and the L_2 key-norm heuristic use the same optimized per-head budgets as every other method and need no calibration pass. Q-Filters is the best of them at every ratio, but the three top out at a mean of 46.5% at 50× over the seven combinations

that report them, where KV-SURGEON-fast reaches 63.8% on the same seven for about forty more seconds per context, so at this compaction calibration is necessary.

	Accuracy (%)					Pipeline time (s)				
Method	100×	50×	20×	10×	5×	100×	50×	20×	10×	5×
L_2 key-norm	37.0 ±0.3	37.9 ±0.4	41.2 ±0.6	51.1 ±0.3	64.2 ±0.3	0.7	0.6	0.5	0.6	0.6
Q-Filters	**47.1** ±0.7	**52.4** ±0.5	**62.0** ±0.2	**66.5** ±0.3	**69.9** ±0.2	0.5	0.6	0.5	0.5	0.6
StreamingLLM	40.4 ±0.6	44.7 ±0.4	52.0 ±0.4	58.4 ±0.4	64.2 ±0.2	0.7	0.5	0.5	0.8	0.7
KVzip	48.0 ±0.4	49.8 ±0.4	61.6 ±0.4	69.1 ±0.5	70.5 ±0.2	16	17	21	26	40
H2O	60.0 ±0.5	62.3 ±0.4	68.2 ±0.4	70.3 ±0.4	71.3 ±0.2	16	17	20	27	40
OBCache	60.1 ±0.3	62.4 ±0.7	68.3 ±0.6	70.2 ±0.3	71.3 ±0.1	16	17	22	27	41
KVMerger	53.7 ±0.4	57.2 ±0.5	63.7 ±0.4	68.6 ±0.1	70.3 ±0.2	36	42	55	76	106
AM-HAK-fast	56.7 ±0.3	58.1 ±0.5	66.5 ±0.3	70.8 ±0.5	**71.9** ±0.1	15	17	20	28	38
KV-SURGEON-fast	**62.4** ±0.1	**65.9** ±0.4	**68.9** ±0.3	**71.3** ±0.5	**71.9** ±0.2	14	16	22	27	28
AM-HAK	59.9 ±0.2	64.1 ±0.9	69.0 ±0.2	71.5 ±0.2	71.7 ±0.4	35	37	41	61	65
AM-OMP-CS	62.2 ±0.6	65.7 ±0.4	**70.6** ±0.2	**71.8** ±0.3	–	51	84	277	919	–
AM-OMP	62.4 ±0.4	66.4 ±0.2	70.1 ±0.2	–	–	117	314	1700	–	–
LOO	61.5 ±0.5	65.6 ±0.2	70.2 ±0.2	71.4 ±0.2	**72.3** ±0.3	37	38	43	61	66
KV-SURGEON	**63.4** ±0.4	**66.8** ±0.3	70.2 ±0.2	71.6 ±0.5	**72.3** ±0.5	37	41	50	68	91

Table 4: QuALITY with Qwen3-4B. Full-context ceiling $72.1 \pm 0.3\%$.

	Accuracy (%)					Pipeline time (s)				
Method	100×	50×	20×	10×	5×	100×	50×	20×	10×	5×
L_2 key-norm	**38.2** ±0.8	37.5 ±0.4	36.0 ±0.6	51.8 ±0.5	64.8 ±0.8	7.0	4.9	4.9	6.0	5.3
Q-Filters	37.6 ±0.4	**43.6** ±0.4	**56.2** ±0.6	**69.1** ±0.5	**76.5** ±0.4	4.8	4.8	5.1	5.4	5.2
StreamingLLM	34.6 ±0.5	39.4 ±0.7	42.9 ±0.8	55.4 ±0.3	64.2 ±0.4	5.3	5.6	5.2	4.6	5.6
KVzip	31.8 ±1.0	39.9 ±0.8	64.6 ±0.6	75.9 ±0.3	77.7 ±0.4	206	222	278	363	530
H2O	49.6 ±0.6	58.4 ±0.4	71.2 ±0.4	76.2 ±0.9	78.5 ±0.2	203	222	277	370	529
OBCache	52.8 ±0.6	62.8 ±0.9	71.2 ±0.4	76.2 ±0.4	77.9 ±1.0	222	233	277	371	555
KVMerger	35.2 ±0.4	45.5 ±0.4	65.6 ±0.8	76.2 ±0.6	78.1 ±0.8	406	461	598	772	1122
AM-HAK-fast	43.5 ±0.9	53.5 ±1.3	**73.0** ±0.8	**78.5** ±0.4	**79.3** ±0.5	204	215	263	357	530
KV-SURGEON-fast	**57.5** ±0.8	**66.9** ±0.5	72.5 ±0.6	76.7 ±0.6	78.8 ±0.5	193	220	297	415	628
AM-HAK	53.7 ±0.4	65.0 ±0.8	73.7 ±0.8	77.2 ±0.5	78.3 ±0.4	331	349	409	512	696
AM-OMP-CS	57.8 ±1.0	65.9 ±1.1	74.5 ±0.9	**79.1** ±0.5	–	586	1091	4518	16653	–
AM-OMP	**59.3** ±0.3	65.5 ±0.6	**76.0** ±0.5	–	–	1678	4884	29489	–	–
LOO	53.6 ±0.4	64.2 ±1.2	74.2 ±0.5	77.5 ±0.5	78.5 ±0.5	353	359	426	551	701
KV-SURGEON	59.1 ±0.5	**67.8** ±0.8	74.8 ±0.7	77.8 ±0.7	**78.8** ±0.4	359	387	523	749	1227

Table 5: LongHealth with Qwen3-4B-Instruct-2507. Full-context ceiling $80.5 \pm 0.8\%$.

Method	Accuracy (%)					Pipeline time (s)				
	100×	50×	20×	10×	5×	100×	50×	20×	10×	5×
L_2 key-norm	38.0 ±0.4	40.2 ±0.5	48.1 ±0.5	56.4 ±0.5	60.6 ±0.1	0.5	0.5	0.5	0.5	0.6
Q-Filters	**40.8** ±0.3	**47.7** ±0.8	**56.7** ±1.0	**60.9** ±0.4	**62.5** ±0.3	0.6	0.6	0.6	0.6	0.6
StreamingLLM	38.7 ±0.5	43.2 ±0.4	49.0 ±0.7	54.1 ±0.4	59.2 ±0.2	0.5	0.5	0.6	0.6	0.6
KVzip	46.0 ±0.6	50.5 ±0.6	58.9 ±0.1	64.1 ±0.6	63.4 ±0.5	14	15	18	24	34
H2O	55.7 ±0.5	59.1 ±0.4	63.1 ±0.3	**65.5** ±0.3	64.2 ±0.4	14	15	18	25	34
OBCache	56.9 ±0.4	59.8 ±0.3	63.2 ±0.4	63.7 ±0.1	64.1 ±0.7	15	16	19	25	35
KVMerger	49.4 ±0.6	54.0 ±0.7	59.6 ±0.4	62.4 ±0.3	63.4 ±0.5	32	37	48	62	92
AM-HAK-fast	54.9 ±0.7	58.7 ±0.6	63.0 ±0.4	63.8 ±0.3	**64.5** ±0.4	14	15	18	24	34
KV-SURGEON-fast	**58.5** ±0.6	**61.0** ±0.4	**63.4** ±0.6	63.8 ±0.6	64.3 ±0.8	13	15	20	22	29
AM-HAK	57.0 ±0.7	60.4 ±0.5	**63.2** ±0.4	63.9 ±0.2	**63.8** ±0.2	28	29	33	40	53
AM-OMP-CS	57.1 ±0.6	60.6 ±0.8	62.8 ±0.9	**64.6** ±0.4	–	42	74	283	934	–
AM-OMP	58.6 ±0.5	61.3 ±0.6	63.1 ±0.7	–	–	101	288	1573	–	–
LOO	58.0 ±0.4	61.2 ±0.7	63.1 ±0.3	63.1 ±0.4	63.2 ±0.3	27	28	32	39	51
KV-SURGEON	**60.3** ±0.7	**62.3** ±0.5	**63.2** ±0.6	63.9 ±0.3	63.5 ±0.5	28	31	39	50	68

Table 6: QuALITY with Llama-3.1-8B-Instruct. Full-context ceiling $62.7 \pm 0.5\%$.

Method	Accuracy (%)					Pipeline time (s)				
	100×	50×	20×	10×	5×	100×	50×	20×	10×	5×
L_2 key-norm	33.0 ±1.2	34.1 ±1.6	37.6 ±0.7	**50.5** ±0.3	**62.5** ±0.6	5.6	5.9	5.6	5.6	6.2
Q-Filters	36.0 ±0.8	36.1 ±1.1	39.9 ±0.6	49.5 ±0.8	61.4 ±0.2	7.0	5.7	6.0	6.6	5.7
StreamingLLM	**37.1** ±1.1	**37.2** ±0.8	**42.2** ±1.1	45.4 ±0.5	56.3 ±0.8	5.9	5.2	5.7	5.6	5.7
KVzip	23.3 ±1.1	41.0 ±1.4	63.5 ±0.7	71.3 ±0.4	72.7 ±0.5	173	179	229	302	437
H2O	43.5 ±0.7	50.6 ±0.9	66.2 ±0.6	70.2 ±0.5	73.2 ±0.6	173	189	220	307	437
OBCache	44.7 ±0.9	52.4 ±0.2	66.6 ±0.4	71.5 ±1.0	73.2 ±0.5	193	202	232	326	457
KVMerger	34.5 ±0.8	42.8 ±0.6	57.3 ±0.2	67.8 ±0.6	72.5 ±0.5	337	397	505	643	929
AM-HAK-fast	44.4 ±1.2	53.5 ±1.1	68.0 ±0.5	**73.5** ±1.4	**74.3** ±0.3	165	180	221	303	447
KV-SURGEON-fast	**53.9** ±1.0	**61.5** ±0.6	**68.8** ±0.4	71.4 ±0.5	73.0 ±0.5	153	181	262	287	403
AM-HAK	48.4 ±1.1	58.2 ±0.5	70.2 ±0.7	72.8 ±0.3	70.5 ±1.0	291	311	350	456	614
AM-OMP-CS	47.9 ±0.5	55.8 ±0.9	66.1 ±0.5	–	–	564	1209	5867	–	–
AM-OMP	49.2 ±0.6	57.3 ±0.9	69.0 ±0.6	–	–	1650	5292	35590	–	–
LOO	44.5 ±1.7	59.1 ±0.6	66.8 ±0.7	70.8 ±0.9	73.4 ±0.6	295	323	360	453	620
KV-SURGEON	**54.0** ±0.8	**62.1** ±0.3	**71.5** ±1.2	**73.2** ±0.5	**74.1** ±0.8	313	349	487	654	908

Table 7: LongHealth with Llama-3.1-8B-Instruct. Full-context ceiling $73.0 \pm 0.8\%$.

Method	Accuracy (%)					Pipeline time (s)				
	100×	50×	20×	10×	5×	100×	50×	20×	10×	5×
L_2 key-norm	41.3 ±0.4	42.5 ±0.6	45.9 ±0.4	52.4 ±0.2	68.3 ±0.3	0.3	0.3	0.3	0.3	0.4
Q-Filters	**46.3** ±0.6	**50.9** ±0.3	**59.9** ±0.7	**70.6** ±0.4	**77.3** ±0.1	0.3	0.4	0.4	0.4	0.3
StreamingLLM	44.7 ±0.5	47.5 ±0.3	53.8 ±0.1	60.9 ±0.4	70.2 ±0.3	0.3	0.3	0.3	0.3	0.3
KVzip	48.3 ±0.2	53.0 ±0.4	60.4 ±0.3	70.3 ±0.7	78.7 ±0.2	56	63	66	96	78
H2O	60.6 ±0.4	66.5 ±0.3	73.7 ±0.7	78.3 ±0.2	**80.5** ±0.4	53	53	59	64	76
OBCache	61.7 ±0.2	67.3 ±0.4	73.1 ±0.5	**78.4** ±0.3	80.0 ±0.3	62	63	61	67	88
KVMerger	58.2 ±0.6	63.9 ±0.6	72.2 ±0.2	75.1 ±0.3	79.2 ±0.1	76	82	94	100	122
AM-HAK-fast	60.3 ±0.3	65.8 ±0.5	71.2 ±0.3	77.3 ±0.4	79.5 ±0.3	51	52	56	63	77
KV-SURGEON-fast	**66.5** ±0.2	**71.4** ±0.3	**76.5** ±0.2	**78.4** ±0.4	80.2 ±0.3	53	54	61	65	76
AM-HAK	63.5 ±0.5	69.5 ±0.3	74.0 ±0.3	78.2 ±0.3	80.3 ±0.4	156	162	168	172	194
AM-OMP-CS	63.5 ±0.6	69.7 ±0.5	75.2 ±0.4	78.5 ±0.4	–	165	183	283	616	–
AM-OMP	63.6 ±0.6	69.2 ±0.4	75.5 ±0.2	–	–	199	300	996	–	–
LOO	64.7 ±0.3	69.6 ±0.3	74.9 ±0.3	78.3 ±0.7	79.8 ±0.1	163	165	170	173	214
KV-SURGEON	**67.6** ±0.4	**72.5** ±0.6	**77.4** ±0.3	**79.2** ±0.2	**80.5** ±0.3	164	159	166	181	198

Table 8: QuALITY with Qwen3-30B-A3B. Full-context ceiling $80.7 \pm 0.4\%$.

Method	Accuracy (%)					Pipeline time (s)				
	100×	50×	20×	10×	5×	100×	50×	20×	10×	5×
OBCache	52.7 ±0.4	62.8 ±0.4	68.6 ±0.5	74.4 ±0.1	76.7 ±0.3	391	395	402	409	424
AM-HAK-fast	50.4 ±1.1	59.8 ±0.6	66.9 ±0.4	74.3 ±0.5	**77.2** ±0.2	358	361	362	372	389
KV-SURGEON-fast	**54.7** ±0.3	**64.1** ±0.2	**70.2** ±0.5	**75.0** ±0.3	77.1 ±0.2	363	366	373	386	413
AM-HAK	51.4 ±0.4	61.4 ±0.4	69.8 ±0.5	73.7 ±0.1	76.8 ±0.3	1121	1129	1130	1159	1162
AM-OMP-CS	52.6 ±0.5	63.1 ±0.2	70.0 ±0.3	–	–	1271	1569	3521	–	–
AM-OMP	53.3 ±0.2	63.9 ±0.6	–	–	–	2147	4011	–	–	–
LOO	52.1 ±0.3	62.3 ±0.7	69.3 ±0.4	74.2 ±1.2	77.0 ±0.3	1109	1112	1131	1143	1159
KV-SURGEON	**55.9** ±0.3	**65.2** ±0.4	**71.4** ±0.4	**75.8** ±0.6	**77.6** ±0.3	1124	1135	1149	1177	1211

Table 9: LongHealth with Qwen3-30B-A3B. Full-context ceiling $78.3 \pm 0.5\%$.

	Accuracy (%)					Pipeline time (s)				
Method	100×	50×	20×	10×	5×	100×	50×	20×	10×	5×
L_2 key-norm	47.4 ±0.5	48.5 ±0.3	49.3 ±0.2	50.8 ±0.1	55.3 ±0.5	0.3	0.3	0.3	0.3	0.3
Q-Filters	**53.2** ±0.5	**55.9** ±0.3	**61.7** ±0.4	**65.4** ±0.4	**68.2** ±0.4	0.3	0.3	0.3	0.3	0.3
StreamingLLM	49.3 ±0.6	50.2 ±0.3	52.4 ±0.5	55.7 ±0.3	61.4 ±0.4	0.3	0.3	0.3	0.3	0.3
KVzip	56.0 ±0.2	57.6 ±0.8	63.1 ±0.4	67.2 ±0.5	69.1 ±0.4	6.0	6.2	6.7	7.4	9.1
H2O	61.5 ±0.3	66.2 ±0.2	68.1 ±0.2	**71.1** ±0.4	**71.3** ±0.4	6.0	6.1	6.6	7.4	9.0
OBCache	62.7 ±0.3	66.6 ±0.4	68.0 ±0.1	70.5 ±0.4	71.2 ±0.2	6.19	6.47	6.84	7.54	9.39
KVMerger	56.5 ±0.2	61.1 ±0.2	64.8 ±0.4	67.6 ±0.3	69.3 ±0.2	12	12	15	19	25
AM-HAK-fast	59.5 ±0.2	62.2 ±0.4	66.8 ±0.5	69.3 ±0.4	69.5 ±0.3	6.1	6.2	6.7	7.4	8.9
KV-SURGEON-fast	**65.2** ±0.3	**68.1** ±0.5	**70.4** ±0.8	71.0 ±0.6	70.6 ±0.2	6.0	6.5	7.4	8.8	11
AM-HAK	61.5 ±0.6	65.1 ±0.3	68.8 ±0.3	70.0 ±0.3	70.9 ±0.4	29	30	30	31	34
AM-OMP-CS	64.2 ±0.3	67.2 ±0.2	69.2 ±0.3	70.3 ±0.3	–	32	37	56	125	–
AM-OMP	64.7 ±0.4	**67.7** ±0.3	69.8 ±0.4	–	–	38	59	180	–	–
LOO	60.5 ±0.3	65.1 ±0.5	68.6 ±0.3	70.5 ±0.4	70.4 ±0.4	29	29	30	31	33
KV-SURGEON	**65.5** ±0.2	67.5 ±0.7	**70.0** ±0.4	**70.9** ±0.4	**71.1** ±0.2	30	31	33	35	39

Table 10: QuALITY with Gemma-3-12B-it. Full-context ceiling $70.6 \pm 0.4\%$.

C.1 LONG CODE ARENA

Table 12 reports project-level code completion on Long Code Arena (Bogomolov et al., 2024) with Qwen3-4B, the experiment behind the last paragraph of Section 4.2. The benchmark is the `large_context` configuration of its project-level code-completion task: each datapoint is a repository snapshot at a fixed commit, one held-out completion file, and a list of target lines in that file.

Contexts and questions. We walk the test split in order and keep the first snapshot of each not-yet-seen repository whose Python files plus completion file fit $120{,}000$ Gemma-3 tokens, until 20 repositories are collected; the dataset order and commit hashes are recorded, so the selection is deterministic. The context of a repository is the concatenation of its non-empty `.py` files in snapshot order, each preceded by a header with its path; the completion file itself is not part of the context. The median context is $73{,}298$ Qwen tokens (range about 45k to 95k). Every listed target line with non-blank content is one question, $n = 650$ in total. A question shows the path of the completion file and its prefix, all lines before the target, in a code fence, and asks for exactly the next line; prefixes longer than $16{,}000$ characters are truncated to their tail at a line boundary, which affects two questions. The prefix is part of the question, not of the compacted context, so compaction is tested through the repository context alone. The answer is generated inside an open code fence with at most 64 new tokens.

Compaction and scoring. As for LongHealth (Appendix G), each repository is compacted in chunks: consecutive files are packed into chunks of at most $10{,}000$ tokens without crossing file boundaries, about nine per repository, each chunk is compacted at the target ratio, and the compacted chunks are concatenated. Head budgets, calibration, final refit, method configurations, and the five-seed protocol are identical to the LongHealth cells; seeds affect calibration-query sampling and generation, not the fixed set of repositories and questions. A completion is correct when its first generated code line, after dropping empty lines and fence lines and stripping surrounding whitespace, matches the reference line exactly; accuracy is the mean over the 650 questions. Time is query generation plus compaction per repository. The full-context ceiling under the same questions and scorer is 40.6%, and the query-free baselines reach 22 to 23% from the file prefix alone, so the effective range of the benchmark is about 22 to 41 points.

Results. At $100\times$ no calibrated method is more than 0.4 points above the best query-free baseline, so the budget rather than the selection rule limits accuracy there. From $50\times$ on, the calibrated methods separate. KV-SURGEON and KV-SURGEON-fast match AM-OMP within uncertainty at $50\times$ at $9.7\times$ and $21.5\times$ less pipeline time, KV-SURGEON has the highest point estimate at $20\times$ and $10\times$, and at $5\times$ every calibrated method except KVzip and KVMerger is within 1.8 points of the ceiling. OBCache is again the strongest ported method, and AM-OMP-CS is the only prior method within uncertainty of KV-SURGEON at $20\times$, at $6.7\times$ its cost.

Method	Accuracy (%)					Pipeline time (s)				
	100×	50×	20×	10×	5×	100×	50×	20×	10×	5×
L_2 key-norm	37.6 ±0.5	38.9 ±1.2	37.8 ±0.6	40.4 ±0.4	41.0 ±0.6	6.2	6.2	6.3	6.1	6.0
Q-Filters	38.4 ±0.2	39.0 ±0.7	**39.4** ±0.9	**44.5** ±0.8	**53.9** ±0.7	6.0	6.1	6.3	6.1	6.1
StreamingLLM	**40.9** ±0.3	**39.5** ±0.9	39.3 ±0.6	42.0 ±0.3	44.4 ±0.6	6.1	5.9	6.0	6.1	6.3
KVzip	38.1 ±0.5	39.8 ±0.7	44.5 ±0.6	52.9 ±0.8	60.7 ±0.6	55	59	64	74	95
H2O	38.5 ±0.6	47.3 ±0.5	55.1 ±0.2	63.5 ±0.6	68.8 ±0.7	54	60	64	74	91
OBCache	41.3 ±0.5	50.2 ±0.8	56.1 ±0.4	64.0 ±0.8	69.7 ±0.5	59.2	59.9	66.4	75.1	95.3
KVMerger	33.7 ±0.6	36.1 ±0.8	42.5 ±0.6	50.1 ±0.9	59.2 ±0.6	97	106	140	170	239
AM-HAK-fast	41.5 ±1.0	45.6 ±1.2	54.6 ±0.8	61.8 ±0.6	66.0 ±0.7	57	59	80	73	96
KV-SURGEON-fast	**45.6** ±0.6	**52.1** ±0.4	**61.1** ±0.3	**66.5** ±0.7	**71.2** ±1.1	53	59	71	97	138
AM-HAK	44.7 ±0.7	52.7 ±1.4	62.5 ±0.6	67.2 ±0.2	71.0 ±0.5	198	207	210	227	291
AM-OMP-CS	43.6 ±0.9	50.9 ±0.8	61.2 ±0.9	66.2 ±0.7	–	225	287	624	2048	–
AM-OMP	45.5 ±1.2	54.5 ±0.9	61.9 ±1.0	–	–	353	689	2831	–	–
LOO	42.5 ±0.7	49.6 ±0.7	60.6 ±0.6	67.2 ±0.7	69.8 ±0.7	206	205	213	218	249
KV-SURGEON	**48.4** ±1.1	**56.4** ±0.7	**63.0** ±0.7	**68.8** ±0.8	**71.8** ±0.5	194	201	224	263	346

Table 11: LongHealth with Gemma-3-12B-it. Full-context ceiling $74.5 \pm 0.4\%$.

Table 12: Code-completion accuracy (%) and pipeline time vs. compaction ratio on Long Code Arena project-level code completion with Qwen3-4B (20 Python repository snapshots, $n = 650$ next-line completions, exact match of the first generated line). Each entry is the mean over five seeds; the grey number is the standard error over seeds. Time is average query-generation plus compaction wall-clock per repository, in seconds. Full-context ceiling: 40.6%. The best accuracy per ratio is bold; underlined entries are within $2\sqrt{\mathrm{se}^2 + \mathrm{se}_{\mathrm{best}}^2}$ of the best. '–' marks ratios the AM-OMP solvers do not reach.

Method	Accuracy (%)					Time (s)				
	100×	50×	20×	10×	5×	100×	50×	20×	10×	5×
StreamingLLM	22.6 ±0.2	23.3 ±0.2	22.0 ±0.1	24.5 ±0.1	32.5 ±0.1	7.62	8.07	7.78	8.18	8.39
L_2 key-norm	23.2 ±0.1	23.4 ±0.1	23.4 ±0.2	23.4 ±0.1	27.6 ±0.2	7.45	8.55	8.77	7.70	8.12
Q-Filters	22.8 ±0.1	22.4 ±0.2	23.8 ±0.1	27.4 ±0.1	34.7 ±0.2	7.11	7.41	7.50	8.02	7.82
KVzip	20.5 ±0.2	22.4 ±0.2	28.6 ±0.4	36.2 ±0.5	38.3 ±0.2	217	232	292	379	596
H2O	22.5 ±0.1	24.1 ±0.3	30.9 ±0.2	36.4 ±0.2	39.4 ±0.2	221	230	307	385	572
KVMerger	13.5 ±0.2	15.1 ±0.2	24.4 ±0.1	31.7 ±0.2	35.9 ±0.2	490	554	688	962	1293
OBCache	22.0 ±0.2	25.4 ±0.3	32.7 ±0.2	37.2 ±0.2	39.2 ±0.3	235	243	309	391	554
AM-HAK-fast	22.2 ±0.4	23.7 ±0.3	32.4 ±0.1	37.4 ±0.1	39.1 ±0.3	222	245	287	384	536
AM-HAK	22.7 ±0.3	24.6 ±0.5	31.0 ±0.2	37.2 ±0.3	39.1 ±0.2	437	448	505	628	846
AM-OMP-CS	23.1 ±0.4	25.5 ±0.3	33.3 ±0.3	–	–	649	1110	5128	–	–
AM-OMP	23.5 ±0.3	**27.0** ±0.4	–	–	–	1646	5037	–	–	–
LOO	22.6 ±0.3	25.3 ±0.2	32.5 ±0.2	37.2 ±0.2	38.8 ±0.4	444	469	534	661	872
KV-SURGEON-fast	22.8 ±0.2	26.5 ±0.2	33.1 ±0.3	36.9 ±0.2	**39.8** ±0.1	204	234	297	407	542
KV-SURGEON	**23.6** ±0.3	26.8 ±0.4	**33.9** ±0.2	**38.1** ±0.3	39.7 ±0.2	474	519	771	875	1322

D COMPONENT ABLATIONS

This section holds every ablation of the shared pipeline and of KV-SURGEON behind Section 4.3 and Table 2: the shared pipeline components in Appendix D.1, compensation in Appendix D.2, and the efficiency and design choices in Appendix D.3. Every ablation runs on Qwen3-4B with QuALITY at $50\times$, changes one component of the pipeline of Section 3 at a time, and reports the mean over five seeds with one standard error over seeds. Two protocols are used and must not be mixed. The 50-article protocol (Tables 13 and 15) is that of the main tables, 894 questions; its self-study controls are the KV-SURGEON, LOO, and AM-HAK entries of Table 1, and its time is pipeline time, calibration-query generation plus compaction. The 25-article protocol (Table 14) uses the first 25 articles, so its accuracies are compared only against the reference row of the same panel and never against the main-table numbers, and its time is compaction only, because query generation is identical across arms. The fast-tier row of Table 13 was run with the fast configuration in use at the time of the ablation, whose reference values (65.5%, 5.4 s) differ from the KV-SURGEON-fast entry of Table 1; it too is read only against its own control. Defaults are those of Appendix B.

D.1 SHARED PIPELINE COMPONENTS: HEAD BUDGETS AND FINAL REFIT

Both components are inherited from attention matching (Zweiger et al., 2026) and applied to every calibrated method, so their effect is measured per method. Table 15 switches them independently for our methods and the attention-matching baselines; the ported eviction methods are walked through the same switches in Table 16.

Head budgets. The ablation replaces the optimized per-head budgets of Appendix B by uniform per-layer budgets at the same global ratio. It is the largest single lever, and it is not specific to KV-SURGEON: with the refit in place, KV-SURGEON loses 6.6 points, AM-HAK 6.9, and LOO 7.8. Under uniform budgets KV-SURGEON remains 2.4 points ahead of LOO and 3.0 ahead of AM-HAK, so the selection-rule margins of Section 4.3 sit on top of a level playing field rather than substituting for one. KVzip, the one baseline that sets its own head allocation from a global attention threshold, points the same way: ported with the shared budgets and refit it collapses under strong compaction, a median of 15.3 points below KV-SURGEON-fast at $50\times$ and $100\times$, yet sits a median of 1.6 points below the ceiling at $5\times$ (Appendix C); the deficit tracks how hard the compaction is, not how good the score is.

Final refit. The refit is ablated on the LOO support, where it can be switched off without touching selection (Table 13). Serving the selected keys with their original gates and values, which is plain eviction on the score of (10), costs 5.4 points against LOO with the refit. Across methods the two switches compound: the budgets are the larger lever, worth $+5$ to $+8$ points on top of the refit, and the refit adds $+1$ to $+7$ on top of raw compaction (Tables 15 and 16). The improvements are not the same for all methods, but KV-SURGEON is ahead in all three settings. Panel (c) of Table 14 decomposes the refit into its two solves on the KV-SURGEON support; it is kept as supporting evidence for Appendix G.1 and is not built on elsewhere.

D.2 KV-SURGEON-SPECIFIC COMPONENT: COMPENSATION

Panel (b) of Table 14 isolates compensation. The surrogate, the $4t$ candidate pool, and the final refit are held fixed and only the compensating update of (7) is disabled, so that elimination proceeds greedily on the same surrogate saliencies. This costs 2.9 ± 0.8 points (seed-paired; 3.0 between the seed means of Table 2), and leaves greedy elimination no better than LOO selection on the same articles (64.7%). Two controls show that the gain is not a better standalone objective. Greedy elimination on the exact set loss of (2) instead of the surrogate reaches 65.8%, 1.1 ± 1.3 below KV-SURGEON and too noisy on 25 articles to separate from either side. Shrinking the candidate pool to the support itself (pool factor 1 in panel (a)), which leaves the elimination nothing to eliminate or compensate, costs the same 2.9 points. The loss-level counterpart, that the compensated support moves toward an exact-refit oracle and that the compensation generalizes to held-out queries, is in Appendix E.

D.3 EFFICIENCY AND DESIGN CHOICES

The remaining choices change accuracy by at most one standard error at their defaults and mainly control cost.

Mass anchor. Setting $\gamma_0 = 0$ removes the anchor from the surrogate (Table 13). Accuracy moves by $+0.2$ points, within one standard error, at 2 s more pipeline time.

Candidate pool. The elimination runs on the $\lceil 4t \rceil$ best entries per head by the score of (10) (panel (a) of Table 14). Widening the pool to the full cache changes accuracy by 0.2 points (66.9 against 66.7) and multiplies compaction time by 3.8 (10 against 38 s); halving it to $2t$ costs 1.0 point.

Plain-eviction fallback. Heads whose budget t exceeds 30% of their cache are served by plain eviction on the score of (10), without pool, elimination, or final refit: eviction is near-lossless there, the $O(t^3)$ value solve is the dominant cost and is largest exactly for these heads, and as t approaches the number of fit rows the solve becomes underdetermined and hurts. Raising the threshold to 1.0 disables the fallback (Table 13). The accuracy of KV-SURGEON-fast is unchanged to the reported precision, and its pipeline time rises from 5.4 to 7.4 s, so the fallback saves 27% of the pipeline. The fallback was not ablated on the self-study tier.

Surrogate refresh. The frozen surrogate can be re-solved at the current retained mass every 256 or 64 eliminations, either reweighting the decoupled residual (panel (a) of Table 14) or consistently, holding that mass fixed so that the refreshed objective agrees with the served output (panel (b)). At the default pool the reweighted refresh costs 1.6 and 1.9 points at $1.3\times$ and $2.2\times$ the compaction time, and the consistent refresh costs 1.3 ± 0.7 and 0.7 ± 0.2 points. Across the pool sweep the refreshed arm is never better than the frozen one beyond one standard error of the difference (the largest apparent gain is $+1.0$ at pool factor 1, with standard errors of 1.0 and 0.7), while its cost grows to $11\times$ at the full cache. At the loss level, measured as in Appendix E, the refresh changes which positions survive, Jaccard 0.55 to 0.85 against the frozen support, without changing the exact loss after the final refit on calibration or held-out queries (within $\pm 2\%$ on every head where a refresh occurs). Freezing is therefore kept for cost, not because it regularizes.

Ablation	Method	Default Acc. (%)	Default Time (s)	Ablated Acc. (%)	Ablated Time (s)
Final refit off (plain eviction on the selected support)					
	LOO	65.6 ±0.2	38	60.2 ±0.4	35
Mass anchor off ($\gamma_0 = 0$)					
	KV-SURGEON	66.8 ±0.3	41	67.0 ±0.4	43
Plain-eviction fallback off (threshold $0.3 \to 1.0$)					
	KV-SURGEON-fast	65.5 ±0.2	5.4	65.5 ±0.2	7.4

Table 13: One-component ablations on Qwen3-4B with QuALITY at $50\times$, all 50 articles, five seeds, one standard error over seeds. Time is pipeline time per context (calibration-query generation plus compaction). The self-study controls are the runs of Table 1; the fast-tier control is the fast configuration in use at the time of the ablation and is read only against its own row. The head-budget ablation is in Table 15.

*(a) Candidate-pool width and reweighted surrogate refresh (*KV-SURGEON*)*

Pool factor	Refresh off Acc. (%)	Refresh off s	Refresh every 256 Acc. (%)	Refresh every 256 s	Refresh every 64 Acc. (%)	Refresh every 64 s
1 (LOO selection)	64.0 ±1.0	7.9	65.0 ±0.7	7.8	64.1 ±0.6	8.0
2	65.9 ±1.0	8.7	66.0 ±0.5	9.2	65.8 ±0.7	10
4 (default)	66.9 ±0.3	10	65.3 ±0.3	13	65.0 ±0.5	22
8	66.5 ±0.7	15	65.5 ±0.4	28	66.0 ±0.3	74
16	66.8 ±0.4	22	66.0 ±0.7	59	64.5 ±0.5	181
50 (full cache)	66.7 ±0.5	38	65.9 ±0.4	126	66.7 ±0.5	416

(b) Selection objective at pool factor 4, refresh off, same final refit

Variant	Acc. (%)	Δ vs. KV-SURGEON
surrogate with compensation (KV-SURGEON)	66.9 ±0.6	—
surrogate, greedy, no compensation	63.9 ±1.6	−2.9 ±0.8
exact loss equation 2, greedy, no compensation	65.8 ±2.8	−1.1 ±1.3
LOO selection (same articles)	64.7 ±0.9	−2.1 ±0.4
consistent refresh every 256	65.6 ±1.5	−1.3 ±0.7
consistent refresh every 64	66.1 ±0.4	−0.7 ±0.2

(c) What the final refit re-solves on the KV-SURGEON *support, pool factor* 4

Final refit re-solves	Refresh off Acc. (%)	Refresh off s	Refresh every 256 Acc. (%)	Refresh every 256 s
values and gates (default)	66.9 ±0.3	10	65.3 ±0.3	13
values only, gates from the elimination	65.5 ±0.5	9.7	64.2 ±0.5	12
gates only, values from the elimination	52.7 ±0.8	12	42.7 ±1.6	12
none, both from the elimination	64.7 ±1.1	8.5	60.9 ±0.5	11

Table 14: Ablations under the 25-article protocol on Qwen3-4B with QuALITY at $50\times$, five seeds, one standard error over seeds; accuracies are compared only within this table. Time is compaction only. (a) The pool holds the $\lceil \text{pool factor} \times t \rceil$ best entries per head by the score of (10); pool factor 1 is LOO selection with the shared final refit and 50 saturates at the full cache. The reweighted refresh re-solves the surrogate at the current retained mass every 256 or 64 eliminations. (b) Greedy rows eliminate on the stated objective without the compensating update of (7); the exact-loss row evaluates (2) on the current support at every step; the consistent refresh re-solves the surrogate at the current retained mass while holding that mass fixed, so that the refreshed objective agrees with the served output. Differences are seed-paired; run times were not recorded for this panel. (c) Whatever the final refit does not re-solve is kept from the coefficients B_S that the elimination itself leaves on the final support.

Table 15: The two shared pipeline components switched independently, at $50\times$ on Qwen3-4B / QuALITY, 50 articles, five seeds: the support-fixed refit (values and gates) and the optimized per-head budgets. All methods use our calibration queries; the AM-OMP value reconstruction cannot be disabled. LOO with optimized budgets and no refit is the plain-eviction row of Table 13; the ported eviction methods are in Table 16.

		uniform budgets		optimized budgets
Method	Queries	no refit	refit	+ refit
AM-HAK	self-study	54.1 ± 0.2	57.2 ± 0.5	64.1 ± 0.9
LOO	self-study	53.2 ± 0.5	57.8 ± 0.6	65.6 ± 0.2
KV-SURGEON	self-study	58.2 ± 0.4	60.2 ± 0.3	66.8 ± 0.3
AM-OMP	self-study	–	59.0 ± 0.5	66.4 ± 0.2
AM-HAK-fast	repeat	48.1 ± 0.5	53.2 ± 0.3	58.1 ± 0.5
KV-SURGEON-fast	repeat	56.3 ± 0.5	58.6 ± 0.5	65.9 ± 0.4
AM-OMP-CS	self-study	–	58.1 ± 0.4	65.7 ± 0.4

E COMPENSATION AT THE LOSS LEVEL AND CALIBRATION ROBUSTNESS

This section measures compensation directly on the attention-output loss rather than through accuracy, and then asks why repeat-pass calibration costs KV-SURGEON less than it costs the exact single-deletion score; Sections 4.3 and 4.4 cite the results. All measurements use Qwen3-4B and Llama-3.1-8B, five QuALITY articles, and every head, that is, 1440 head–article pairs on Qwen3-4B and 1280 on Llama-3.1-8B. Scores are computed from the self-study calibration queries, and the exact loss of (2) is evaluated on held-out benchmark questions the scores never saw; where the same measurement on the calibration queries differs, it is given in parentheses.

Compensation moves the support toward the oracle and generalizes. Against an exact-refit oracle on the 546 heads small enough to enumerate ($t \leq 57$, Qwen3-4B), LOO selection sits 46% above the oracle's loss and KV-SURGEON 25% (median relative loss 0.213, 0.183, and 0.146; random selection 0.243), closing 45% of the gap, and on the typical head the closed fraction is 39%. KV-SURGEON beats LOO on 84% of heads and still shares only 44% of its keys with the oracle set, because many near-equivalent supports exist. On the supports KV-SURGEON selects at $50\times$, the final refit leaves 25% of the raw-eviction loss on Qwen and 7.5% on Llama on held-out questions, against 25% and 7.7% on the calibration queries: no head overfits, and on no head is the stored cache worse than eviction. On Llama raw eviction at $50\times$ leaves a loss larger than the output itself, so compensation rather than selection is what makes the method work there. The coefficients the elimination itself produces already achieve most of this (31% and 8%), but they cannot be served raw: on 1% of Qwen heads and 5% of Llama heads they exceed the eviction loss, and the final refit never does, which is the loss-level counterpart of the final-refit ablation of Appendix D.1.

Why repeat-pass calibration costs KV-SURGEON less than the exact score. Averaged over the eight combinations at $50\times$, replacing self-study queries with a single repeat pass costs KV-SURGEON 1.2 points (Table 1); the drop is 4.3 points on LongHealth with Gemma-3-12B and at most 1.3 elsewhere. At the level of rankings, recomputing the scores from a single repeat pass drops the Spearman correlation between the surrogate and the exact loss over random deletion sets, computed within one head and one set size, from 0.905 to 0.786 on Qwen and from 0.929 to 0.833 on Llama, and that of the score of (10) from 0.881 to 0.690 on both, so under cheap queries the surrogate loses about half the ranking fidelity the exact score loses. Cross-head comparability, which is what the per-head budget allocation consumes, is unaffected (0.81 and 0.91 at 98% deletion), so budget allocation does not need self-study queries. This is a halving of the damage, not immunity.

Hypothesis, not tested as a mechanism. The factor $(1 - P_{ij})^{-2}$ in the exact deletion score amplifies exactly the queries a repeat pass gets wrong: a query for which one token receives unusually large attention contributes disproportionately to that token's score, so an atypical repeat-pass attention pattern moves the exact ranking more than it moves the frozen surrogate, which contains no such amplification. This is consistent with the measurements above and is a candidate mechanism for the fast-tier robustness of Section 4.4, but it does not establish why AM-HAK degrades more strongly under the same change: AM-HAK ranks tokens by attention mass rather than by either of these objectives, so its larger calibration drop may have a different cause.

F PORTED BASELINES: FROM THE PUBLISHED REGIME TO OURS

H2O, OBCache, KVMerger, and KVzip are serving-time or self-allocating methods; Section 4.1 describes how each is ported into the offline pipeline. Each port replaces the method's own queries by the shared repeat-pass calibration queries, keeps its selection score, and serves the selected support with the shared per-head budgets and final refit of Section 3.4. For OBCache, the published observation window of the last 64 prompt tokens is replaced by the calibration queries, and its second-order score is aggregated over them. Table 16 validates the port by walking each method from its published configuration to our pipeline one component at a time, at $50\times$ on Qwen3-4B with QuALITY under the 50-article protocol of Appendix B.

Table 16: Each external baseline from its published configuration to our pipeline, one component at a time, at $50\times$ on Qwen3-4B / QuALITY. As published: own queries (H2O and KVMerger use context prefill, OBCache its last-64 observation window, KVZip its repeat prompt with a global cross-head threshold), uniform or own head allocation, no refit. The remaining columns use repeat-pass calibration queries.

Method	as published	calib. queries	+ refit	+ budgets
H2O	47.4 ± 0.5	48.7 ± 0.4	56.1 ± 0.4	62.3 ± 0.4
OBCache	43.2 ± 0.2	53.0 ± 0.4	58.1 ± 0.3	62.4 ± 0.7
KVMerger	47.7 ± 0.5	47.6 ± 0.2	51.5 ± 0.5	57.2 ± 0.5
KVZip	36.4 ± 0.3	42.6 ± 0.3	43.5 ± 0.6	49.8 ± 0.4

The pipeline we use, inherited from Zweiger et al. (2026), is worth 9 to 19 points to each of the eviction methods we ported to the offline compaction regime; the budgets are the largest single step for KVMerger and KVzip, the refit for H2O, and the calibration queries for OBCache. Even with every component, the best ported rule, OBCache at 62.4, stays 3.5 points below KV-SURGEON-fast (65.9) at the same cost, and H2O at 62.3 is within noise of OBCache. The two hybrid configurations with their own queries but our budgets (H2O 58.7 ± 0.2, KVMerger 57.5 ± 0.5) confirm that budgets alone do not close the gap.

G CACHE CONSTRUCTION AND THE ROLE OF MASS MATCHING

Chunked contexts. Long structured contexts, for example records of several patients, are compacted one chunk at a time. At inference the compacted blocks sit side by side in one cache, and the softmax normalizes across all of them. Block b therefore competes for the attention of query i through its total unnormalized mass $m_i^b = \sum_{j \in S_b} w_j e^{z_{ij}}$. A compaction that shrinks a block's mass re-routes attention away from that block and toward its neighbors, even when the reconstruction within the block is perfect. In this setting the absolute mass of a block is meaningful, and it must be preserved alongside the outputs (Section G.1).

G.1 THE ROLE AND LIMITS OF MASS MATCHING

Attention matching (Zweiger et al., 2026), the strongest prior offline method, builds its selection and refit around reproducing each query's attention *mass* on the compacted block. Two consequences of Section 3.2 matter here: for how the refit must be run, and for what mass matching can tell selection.

Per-position mass fitting is harmful alone and misleading as a selection signal. Non-uniform gates change the retained attention pattern, so the original values no longer reconstruct o_i under the reweighted pattern. Fitting gates to mass targets without the subsequent value solve is therefore worse than no refit at all, which is why the refit of Section 3.1 applies the two solves only as an ordered unit. This is measured on the KV-SURGEON support: re-solving the gates alone scores 12.0 points below serving the elimination's own coefficients with no refit at all, and 14.2 below the full final refit (panel (c) of Table 14). The mechanism is the reweighting rather than the mass objective in particular, so the ablation bounds what mass fitting costs when it is run alone and does not separate it from other gate objectives. Mass matching also misleads selection. Weighting the mass channel heavily, formally $\gamma \to \infty$ in the anchor, ranks positions by the attention they receive, which is the criterion of AM-HAK. This discards both factors that the exact loss equation 10 says matter, the value novelty $\|v_j - o_i\|^2$ and the heavy-hitter weight $(1 - P_{ij})^{-2}$. At mild compaction the difference is small, because retained masses stay near one and all criteria agree. At aggressive compaction the two criteria part ways, and the exact loss sides with the output: replacing mass ranking by KV-SURGEON selection inside an otherwise identical pipeline is worth 1.3 points at $20\times$, 3.0 at $50\times$, and 4.3 at $100\times$ (Figure 2).

None of this contradicts the published attention-matching results, because all of those results combine mass matching with a value refit.

H GATE DIAGNOSTICS FOR THE UNCONSTRAINED ELIMINATION

Table 17: Diagnostics of the unconstrained OBS gates w and the served gates b after the box-constrained refit of (8), per ratio, on Qwen3-4B / QuALITY (seed 0, all 14,400 head instances; heads served by fallback or zero budget excluded). Columns: fraction of kept gates with $w < 0$ (gate-weighted, and the median head); fraction of heads with at least one negative gate; median per-head relative movement $\|b - w\|_2 / \|w\|_2$ and correlation of b against w; fraction of served gates pinned at the bounds $e^{\pm\beta}$, $\beta = 3$.

Ratio	$w < 0$	median head	heads w/ any	rel. move	corr.	at bounds
$100\times$	0.42%	0.00%	13.6%	0.81	0.39	1.3% / 14.7%
$50\times$	0.33%	0.00%	19.4%	0.70	0.51	1.3% / 9.4%
$20\times$	0.24%	0.00%	24.6%	0.57	0.62	1.8% / 5.9%
$10\times$	0.24%	0.00%	28.0%	0.54	0.68	2.9% / 5.5%
$5\times$	0.28%	0.00%	33.0%	0.58	0.70	4.7% / 4.1%

Table 17 shows the impact of a relaxation that makes closed-form compensation possible, i.e., relaxing the gates to real numbers instead of positive numbers. Fewer than half a percent of kept gates end negative at any ratio, the median head has no negative weights, and the numbers are unchanged between the repeat and self-study query tiers. The refit is calibration, not sign repair: it rescales substantially (median relative movement 0.5 to 0.8, correlation 0.4 to 0.7) and its pinning at the

upper bound grows from 4% of gates at $5\times$ to 15% at $100\times$, which is where the mass anchor works. Negative gates concentrate in early layers (1–6) and refit movement is largest in late layers (25–34).