# BELLOWSKV: ONE LATENT FOR MANY LAYERS, EXPANDED ON DEMAND

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Large language models (LLMs) are now widely utilized as the backbone of agents that reason over many turns, call tools, and pursue long-horizon tasks. To serve such a context without recomputing the past, a model stores the attention keys and values of all past tokens in a key-value (KV) cache. Each layer maintains its own cache and re-reads it at every generation step, so KV-cache memory can become the main bottleneck for GPU serving. Most compression methods operate within a single layer, by quantizing the cache entries, reducing the channel dimension, or evicting less important tokens from the cache. All three lose substantial accuracy at high compression ratios. A more recent line of work shares one compressed cache across adjacent layers to go beyond this limit. However, existing cross-layer methods also degrade sharply at high compression ratios. We find that the representation of KV Cache between adjacent layers are quite different, whereas the input hidden states are highly similar. We also find that, under the same memory budget, reducing the dimension of this shared latent loses far more accuracy than quantizing it. Based on these findings, we propose BellowsKV, a training-free method that maps the hidden states of n adjacent layers to one shared latent per token. At inference time, each layer *read* its own keys and values from this latent through a layer-specific projection. We keep the latent at full dimension and compress it in the two other ways instead: BellowsKV-Quant stores the latent at low precision, which is nearly lossless because the latent has no dominant outlier channels; BellowsKV-Evict makes one eviction decision per block, which the layers can share because they rank token importance very similarly. Across Llama-3.1-8B, Mistral-7B and Qwen2.5-7B, BellowsKV-Quant retains about 99% of full-cache accuracy at 3.6× compression and 97-98% at 13-14×, where the strongest existing cross-layer method retains at most 84% at 16×.

## 1 INTRODUCTION

The way large language models are used has shifted from answering questions to carrying out tasks. Agents debug a repository over hundreds of tool calls with the accumulated history in view (Yang et al., 2024a; Jimenez et al., 2024), assistants read and cross-reference whole document collections in one pass (Bai et al., 2024; Hsieh et al., 2024), and reasoning models think in traces tens of thousands of tokens long before they answer (OpenAI et al., 2026; Guo et al., 2025). Context windows have chased these workloads, from the 4K to 128K as the open-weight default (Grattafiori et al., 2024) and past a million tokens at the frontier (Team et al., 2024; Yang et al., 2025a). The key-value (KV) cache holds the attention keys and values of every token in the window, and each layer keeps its own cache. This way, we do not need to recompute past tokens, because we can read the cache in full at every decoding step. At this scale the cache stops being a rounding error on the weights and becomes their rival: on Llama-3.1-8B a single 131K-token sequence carries a 16.8 GB cache in 16-bit precision, slightly more than the model itself, after grouped-query attention has already shrunk it fourfold (Ainslie et al., 2023). In the real system, the LLM weights are loaded once and shared by every request in a batch, but each sequence needs its own KV cache. Serving is economical only when many sequences run at the same time (Kwon et al., 2023), so the total cache size grows with the batch size while the weights stay fixed. A batch of eight 131K-token sequences, for example, holds 134 GB of cache against 16 GB of weights. Compressing it by 4x, where most published methods operate, would still leave 33 GB. This paper targets an order of magnitude instead.



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.

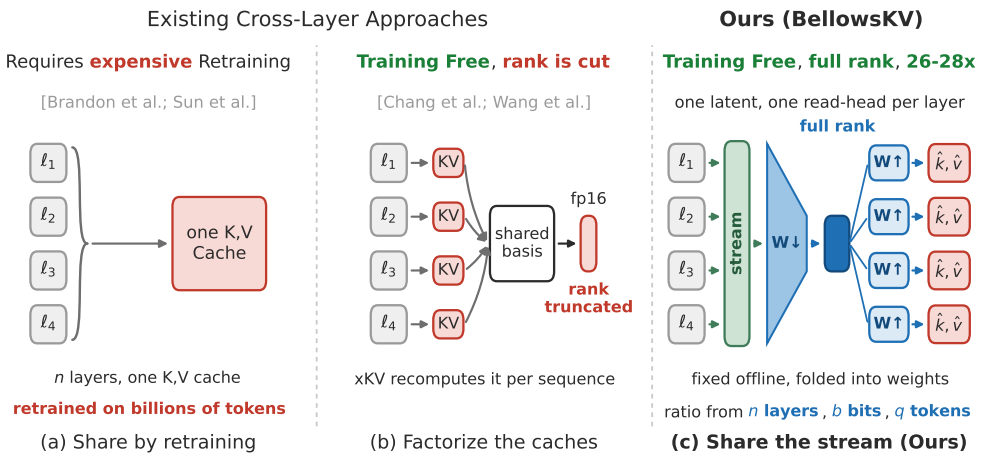



Figure 1: **Comparison of different cross-layer KV-Cache compression methods**: (a) Trained-to-share architectures reach one cache only by retraining. (b) Training-free methods factorize the caches (c) BellowsKV folds the block's hidden states into one latent, buying the compression ratio from the numbers of shared layers, bits and tokens instead.

Nearly all existing KV cache compression methods work inside a single layer. Cached keys and values are quantized to fewer bits (Liu et al., 2024b; Su et al., 2025b; Zandieh et al., 2025; Han et al., 2025), tokens judged unimportant are evicted (Zhang et al., 2023; Li et al., 2024; Xiao et al., 2024), and feature channels are pruned or projected to lower rank (Chang et al., 2024; Saxena et al., 2024; Staniszewski & Łańcucki, 2026). All three directions work at moderate compression ratios, but each of them is close to its limit when further compressing the cache.

Quantization loses accuracy sharply below two bits, and accuracy depends on extra machinery around the quantization methods, such as choosing the axis along which numbers share a scale, rotating the cache so that no channel dominates a scale, or keeping a sparse set of outliers in full precision (Hooper et al., 2024). Token eviction cannot be undone when regret, because a token dropped after prefill is gone for every later decoding step and for every follow-up turn, even if attention would have needed it. Reducing the channel dimension removes components of the keys and values that attention still uses. A recent position paper argues that at a matched memory budget, quantization is preferable to rank reduction (Salfati, 2026).

However, adjacent layers compute their caches from almost the same hidden states, yet every layer still stores its own cache in full. Sharing a cache across layers is therefore the natural next move, and it has recently been tried. xKV (Chang et al., 2026) factorizes the caches of a window of layers into one shared low-rank basis, recomputed for every sequence at prefill. CommonKV (Wang et al., 2025) instead derives a shared projection from the weight matrices of the grouped layers. Both work at 4x compression, but at 16x, xKV retains at most 84% of full-cache accuracy and CommonKV at most 54%. To our knowledge, no prior work explains why cross-layer methods degrade at these ratios, or what a block of layers needs in order to share one cache without losing accuracy. Appendix A reviews all three families in full, together with the architectures that are trained to share a cache.

Therefore, we propose BellowsKV, a training-free KV cache compression method built on these three measurements. For each block of n adjacent layers, we fit a single projection offline, and it maps the hidden states of the block to one latent vector per token, and each layer reconstructs its own keys and values from that latent through its own up-projection. Two operators then compress the latent: BellowsKV-Quant stores it with uniform low-bit quantization, and BellowsKV-Evict drops tokens with a single eviction decision for the whole block. Our contributions are:

- We study what a block of adjacent layers can share. The keys and values of adjacent layers are nearly orthogonal, while the hidden states they are computed from are highly similar. The representation shared by a block tolerates quantization and eviction, but not dimension reduction (Section 2).

- We propose BellowsKV, a training-free method that stores one latent per block of $n$ layers, together with two operators that compress this latent, BellowsKV-Quant and BellowsKV-Evict (Section 3).

## 2 Motivation: What Should a Block of Layers Store?

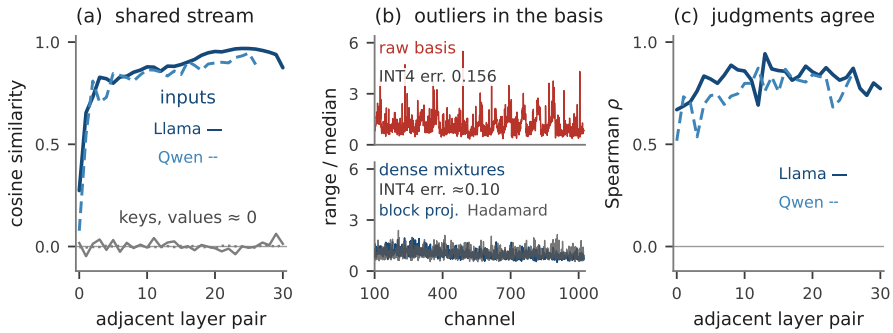



Figure 2: (a) Cosine similarity between adjacent layers. The keys and values are close to zero, while the hidden states reach $0.87$ on Llama and $0.82$ on Qwen. (b) Per-channel dynamic range of the keys. The raw cache concentrates the range in a few channels (top). A data-free Hadamard rotation and our block-level projection are both flat (bottom), and the relative INT4 error falls from $0.156$ to about $0.10$. (c) Spearman correlation between the token importance scores of adjacent layers, between $0.76$ and $0.80$.

Every cache compression method makes two decisions: what to store, and how coarsely to store it. Existing methods only make the second: quantization, eviction and channel reduction keep one private cache per layer, as the architecture defines it, and compress that cache in place. This paper starts from the first decision. We ask what a block of $n$ layers should store, under two requirements: every layer must still recover its own keys and values from the stored object, and the object must remain accurate when stored coarsely. Figure 2 answers with measurements on the uncompressed models, which do not assume any particular compression method.

### 2.1 Adjacent layers have similar hidden states but dissimilar cache

A natural reading of cross-layer redundancy is that a later layer could reuse an earlier layer's cache. Our measurements rule this out. Across adjacent layer pairs of both models, the per-token cosine similarity between neighboring keys is close to zero, and the same holds for values (Figure 2a). Substituting one shared key across even two layers increases perplexity significantly, even when the shared key is chosen by an oracle criteria (Appendix B.9).

The reason is that adjacent layers compute their keys and values from the same input, but with different projection matrices. Therefore, the keys and values of two adjacent layers are not similar to each other. In contrast, the hidden states remain similar across adjacent layers, and the similarity increases with depth (Figure 2a), and the same observation is also reported by Wang et al. (2025), which attribute such observation to the residual connection. These results show that the redundancy between adjacent layers exists in the inputs of the caches, and not in the caches themselves. Several architectures are trained to share one cache across a group of layers (Brandon et al., 2024; Sun et al., 2024; Zuhri et al., 2024; Wu & Tu, 2024). These architectures show that a group of layers can work with a single cache. However, they require training the attention layers on billions of tokens, which is not possible for a model that is already trained. *Therefore, a block of layers should store one representation computed from its hidden states, rather than from its caches.* Each layer then reconstructs its own keys and values from this representation through its own up-projection.

### 2.2 What aggregation removes, and the one thing it cannot

The requirement above fixes the structure of the shared representation. To serve $n$ layers at once, the representation must combine the channels of all $n$ layers into common coordinates, and it must

store one entry per token for the whole block. We now ask how coarsely such a representation can be stored. We consider three forms of compression: quantization (fewer bits per number), eviction (fewer tokens), and dimension reduction (fewer coordinates per token).

**Observation 1: The shared representation has no dominant outlier channels** Uniform quantization fails when a few channels have a much larger dynamic range than the rest of the channels because these channels determine the scale of the group. The KV cache is known to contain such channels (Liu et al., 2024b; Hooper et al., 2024); see Figure 2b (top). A common solution is to rotate the cache before quantization, so that every stored coordinate becomes a mixture of all channels and no single channel determines the scale. However, the rotation must be applied again at every read (Ashkboos et al., 2024; Liu et al., 2025; Su et al., 2025b; Han et al., 2025; Zhou et al., 2026; Su et al., 2026). The shared representation already has this property, because every coordinate it stores is a linear combination of channels from all $n$ layers. To verify this, we construct the block-level representation defined above and quantize it with a uniform quantization. Also, we apply the same uniform quantization to a rotated copy of the raw cache. The two are equally flat, and the quantization error of INT4 precision format is $0.10$ for the shared representation against $0.156$ for the raw cache (Figure 2b, bottom). Therefore, the shared representation does not need a rotation before it is stored.

**Observation 2: The layers of a block agree on which tokens are important** A token can be evicted only if it is unimportant, and each layer decides this from its own attention scores. We therefore compare the importance scores that the layers of a block assign to the same tokens. Adjacent layers rank tokens very similarly, with a Spearman correlation between $0.76$ and $0.80$, and the correlation increases with depth (Figure 2c). The same agreement between neighboring layers is also reported in previous work (Wu et al., 2025; Liu et al., 2024a; Ma et al., 2025). A single eviction decision, computed from the scores of all $n$ layers, can therefore replace the $n$ per-layer decisions.

**Observation 3: The dimension of the shared representation cannot be reduced without a large loss of accuracy** The first two properties follow from mixing the channels of $n$ layers. However, mixing does not reduce the rank of the representation. Therefore, storing fewer coordinates per token removes information that the layers still need at read time. We do not measure this property in this section. We test it directly in Section 4.2, where we compare two ways to reach the same memory budget: reducing the dimension of the representation, and reducing the number of bits per coordinate.

These three observations define the design of BellowsKV.[1] For each block of $n$ layers, we compute one shared representation in closed form from the activation statistics of the model, and each layer reconstructs its own keys and values from this representation when it reads the cache. Observation 1 and Observation 2 lead to the two operators of BellowsKV: BellowsKV-Quant stores the representation with uniform low-bit quantization, and BellowsKV-Evict applies one eviction decision to the whole block (Section 3).

## 3 METHOD

A decoder-only transformer has $L$ layers of attention modules with hidden width $d$. At layer $\ell$ and position $t$ in a sequence of token, attention module reads the hidden state $\mathbf{h}_{\ell,t} \in \mathbb{R}^d$ and forms $\mathbf{k}_{\ell,t} = \mathbf{W}_K^{\ell\top}\mathbf{h}_{\ell,t}$ and $\mathbf{v}_{\ell,t} = \mathbf{W}_V^{\ell\top}\mathbf{h}_{\ell,t}$ with projection matrices $\mathbf{W}_K^{\ell}, \mathbf{W}_V^{\ell} \in \mathbb{R}^{d\times p}$, both cached for every layer and every past position, and for simplicity we write attention as single-headed with key width $p$. Since the construction never looks inside the key vector, the MHA and GQA (Ainslie et al., 2023) follow verbatim. Bold lowercase letters are column vectors and bold uppercase letters are matrices; $[\mathbf{a};\mathbf{b}]$ stacks $\mathbf{a}$ on top of $\mathbf{b}$, and $\|\cdot\|_F$ is the Frobenius norm. Bias terms are present in every computation but omitted from the notation.

The $L$ layers are partitioned into consecutive, non-overlapping blocks of $n$ layers, $\mathcal{B} = (\ell_i, \ell_{i+1}, \ldots, \ell_{i+n-1})$; for example, $n$=3 gives the blocks $(\ell_0, \ell_1, \ell_2)$, $(\ell_3, \ell_4, \ell_5)$, and so on, and a final block shorter than $n$ is kept as is.

The method has two parts. The first part is a projection that is computed once per model, offline, from the activation statistics of the block, and is then held fixed for every sequence. This projection

[1] The name refers to a bellows, which folds a large volume into a small one and expands it again on demand.

maps the hidden states of the block to one latent vector per token, which we call the *consensus*, and it produces all of the compression. The second part is a low-rank correction, which we call the *whisper*, and it is fitted at prefill for each sequence. In the following section, we write the key side only: values have their own latent and up-projections, constructed identically with $\mathbf{W}_V$ in place of $\mathbf{W}_K$, and we mark $K$ and $V$ explicitly only where the two appear together or are treated differently.

### 3.1 One latent per block, one up-projection per layer

For a block $\mathcal{B}$ and token $t$, BellowsKV replaces the block's $n$ cached keys and values with two latents $\mathbf{c}_t^K, \mathbf{c}_t^V \in \mathbb{R}^R$ ($R \ll np$), from which each layer recovers its own key through a fixed up-projection. For example:

$$\mathbf{c}_t = \mathbf{W}_{\downarrow}^{\top}\, \mathbf{h}_{\mathcal{B},t}, \qquad \hat{\mathbf{k}}_{\ell_i,t} = \mathbf{W}_{\uparrow}^{(i)\,\top}\, \mathbf{c}_t, \qquad \mathbf{h}_{\mathcal{B},t} = \left[\mathbf{h}_{\ell_1,t}; \ldots; \mathbf{h}_{\ell_n,t}\right] \in \mathbb{R}^{nd}, \tag{1}$$

where $i \in \{1, \ldots, n\}$ indexes the layers inside the block. The down-projection $\mathbf{W}_{\downarrow} \in \mathbb{R}^{nd \times R}$ is shared by the whole block and applied to the vertical stack $\mathbf{h}_{\mathcal{B},t}$ of its $n$ hidden states. The up-projection $\mathbf{W}_{\uparrow}^{(i)} \in \mathbb{R}^{R \times p}$ belongs to layer $\ell_i$. It maps the shared latent back to $\hat{\mathbf{k}}_{\ell_i,t}$, which is an approximation of the exact key $\mathbf{k}_{\ell_i,t} = \mathbf{W}_K^{\ell_i\,\top}\mathbf{h}_{\ell_i,t}$ of the uncompressed model. All $n$ layers read the same latent $\mathbf{c}_t$, and each layer applies its own up-projection. The values use an independent latent of the same form, $\mathbf{c}_t^V = \mathbf{W}_{\downarrow}^{V\,\top}\mathbf{h}_{\mathcal{B},t}$, with its own up-projections $\mathbf{W}_{\uparrow}^{V,(i)}$. The construction is identical.

The cache therefore holds, per block and per token, the pair $(\mathbf{c}_t^K, \mathbf{c}_t^V)$ of $2R$ numbers in place of $2np$, plus a short per-token whisper correction (Section 3.3). All four factor sets are fixed model parameters, computed once offline (Section 3.2) and folded into the weights, so the compression itself is never recomputed per sequence; the whisper is the one per-sequence addition, and it is small. This is exactly the object Section 2 argued a block should store.

### 3.2 Calibrating the projection and the up-projections

To minimize the reconstruction error between the exact cache and the cache BellowsKV reads back from the latent, we fit the factors on calibration data. To set this up, we stack a calibration batch of $S$ tokens into $\mathbf{H} \in \mathbb{R}^{S \times nd}$ with rows $\mathbf{h}_{\mathcal{B},t}^{\top}$, write $\mathbf{W} = \mathrm{blkdiag}(\mathbf{W}_K^{\ell_1}, \ldots, \mathbf{W}_K^{\ell_n}) \in \mathbb{R}^{nd \times np}$ for the block's true key map, so that the block's exact keys are the rows of $\mathbf{HW}$, and collect the up-projections as $\mathbf{W}_{\uparrow} = [\mathbf{W}_{\uparrow}^{(1)}, \ldots, \mathbf{W}_{\uparrow}^{(n)}] \in \mathbb{R}^{R \times np}$. The objective is

$$J(\mathbf{W}_{\downarrow}, \mathbf{W}_{\uparrow}) = \mathbb{E}_{\mathbf{H}} \left\| \mathbf{HW} - \mathbf{HW}_{\downarrow}\mathbf{W}_{\uparrow} \right\|_F^2. \tag{2}$$

The data enters through one statistic, the second-moment matrix $\mathbf{G} = \mathbb{E}[\mathbf{H}^{\top}\mathbf{H}] \in \mathbb{R}^{nd \times nd}$, whose $(i, j)$ block is the covariance between layer $\ell_i$'s and layer $\ell_j$'s inputs. Eq. (2) has a closed-form solution. Let $\mathbf{M} = \mathbf{G}^{1/2}\mathbf{W}$ with thin singular value decomposition $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}$. The off-diagonal blocks of $\mathbf{G}$ are the part that BellowsKV adds. If these blocks were zero, $\mathbf{G}$ and $\mathbf{M}$ would be block-diagonal, the singular value decomposition would split into $n$ independent per-layer decompositions, and Eq. (2) would reduce to $n$ separate per-layer low-rank caches. This per-layer construction is used by Chang et al. (2024) and Zhang et al. (2026), and it is based on the activation-aware whitening of Wang et al. (2024). With them present, one set of $R$ latent directions is fitted to serve all $n$ layers at once. Then

$$\mathbf{W}_{\downarrow}^{\star} = \mathbf{G}^{-1/2}\,\mathbf{U}_{:R}, \qquad \mathbf{W}_{\uparrow}^{\star} = \mathbf{\Sigma}_{:R}\mathbf{V}_{:R}^{\top}, \tag{3}$$

attains the minimum of Eq. (2) with residual $\sum_{i>R}\sigma_i^2$ (Appendix B). The solution is unique up to an invertible mixing of the latent coordinates, invisible at full precision but not to a quantizer; we take the square-root split $\mathbf{W}_{\downarrow} = \mathbf{G}^{-1/2}\mathbf{U}_{:R}\mathbf{\Sigma}_{:R}^{1/2}$, $\mathbf{W}_{\uparrow} = \mathbf{\Sigma}_{:R}^{1/2}\mathbf{V}_{:R}^{\top}$, which leaves the latent energy-ordered and so matched to the quantizer's group-wise scales.

### 3.3 The cache at inference and the two operators

At prefill, each block runs its layers on the exact keys and values, applies Eq. (1) to the stacked hidden states, and stores $(\mathbf{c}_t^K, \mathbf{c}_t^V)$ in place of the block's $2n$ per-layer tensors; a few sink positions, a

short recent window, and every token generated at decode time stay exact and per-layer, the standard residual-window practice (Xiao et al., 2024; Liu et al., 2024b). Before storing, the block fits its whisper: a low-rank sketch (Halko et al., 2010) of the key residual $\mathbf{E} = \mathbf{K} - \hat{\mathbf{K}}$, one small basis per sequence and a few coefficients per token. At decode, values never leave the latent: attention is linear in them, so each up-projection folds into the output projection and the layer attends in $R$ dimensions (Ji et al., 2025). Keys are reconstructed through theirs, because the positional rotation sits between latent and score (Su et al., 2023; DeepSeek-AI et al., 2024): a low-rank product per layer that hides under the memory-bound cache read (settings and full path in Appendix B).

BellowsKV has three parameters: the number of layers per block $n$, the number of bits per stored value $b$, and the fraction of tokens kept $q$, and the dimension is fixed at $R = p$.

**BellowsKV-Quant.** We quantize the latent with uniform integer quantization, using one scale per group of 128 adjacent coordinates. We apply no rotation before storing (Ashkboos et al., 2024; Su et al., 2025b) and no per-channel treatment, because the latent is already a mixture of channels from all $n$ layers (Observation 1). The dimension stays at $R = p$ in all settings. The main experiments use two settings: $n = 2$ with INT8 and $n = 4$ with INT4, which measure $3.6\times$ and $13.8\times$ with all metadata counted (Appendix B.3).

**BellowsKV-Evict.** The block stores one latent vector per token, so a token receives one eviction decision per block. After prefill, each block scores every cached token by the attention mass that the most recent queries place on it. The score is summed over the layers and heads of the block, and it is computed on the keys reconstructed from the quantized latent, so that eviction uses the same keys as decoding (Zhang et al., 2023; Li et al., 2024). The scores are then max-pooled over a window of 13 positions, so that a token kept for its own score also keeps its neighbors (Li et al., 2024). The tokens with the highest scores are kept, together with the first 4 positions and the last 64 prompt positions (Xiao et al., 2024). The remaining tokens are dropped once for the whole block, with a single gather applied to the stored latent (Appendix B.4).

## 4 EXPERIMENT

We evaluate BellowsKV on Llama-3.1-8B-Instruct, Mistral-7B-Instruct-v0.3, Qwen2.5-7B-Instruct, Qwen3-32B and Qwen3-30B-A3B, with one recipe fixed throughout (Section 3): $n$=2 with INT8 at the low tier and $n$=4 with INT4 at the high tier, no per-model tuning; the high tier is also stored at INT3 and at K3V2 (key latent 3 bits, value latent 2) to reach and pass $16\times$. *Protocol A* reports absolute scores on eight English LongBench tasks (Bai et al., 2024) under the official harness; *Protocol B* reports *retention*, each task's score over the dense model's in the same harness, on 4k–32k-token inputs under the xKV authors' harness (Chang et al., 2026). *Protocal C* reports agent workload: single-turn retrieve-then-read on BrowseComp-Plus Chen et al. (2025).

Every number is full-sample, every ratio counts all metadata, and every table groups rows by measured ratio and marks the best per tier. Tables abbreviate BellowsKV-Quant as **BellowsQ**, BellowsKV-Evict as **BellowsE** and the sharing-only fp16 latent as **BellowsS**. Each subsection tests one prediction of Section 2 where it would fail, and the last tests the decode relation the introduction promised.

### 4.1 STREAM, NOT CACHES: THE CROSS-LAYER HEAD-TO-HEAD

Section 2.1 predicts that reusing one layer's raw keys and values in another fails and storing a latent of the stream does not, at any ratio. Protocol B runs every training-free cross-layer method we know of, placed by measured bytes (Table 1, Appendix Figures 3–5): xKV (Chang et al., 2026) and CommonKV (Wang et al., 2025) store a latent; MiniCache (Liu et al., 2024a), KVSharer (Yang et al., 2024b) and XQuant (Yang et al., 2025b) share raw caches. The raw-cache routes are broken at $1.3\times$, before any bits are cut; every latent method works at $\sim 4\times$; at $\sim 16\times$ xKV retains at most 84% and CommonKV 54%, while BellowsQ retains 97–98% at 13–$14\times$, at INT3 (fewer bytes than xKV's point) and at 19–$20.5\times$, where no cross-layer point exists. The raw-cache methods already lose accuracy at $1.3\times$, which shows that the caches themselves are not the object to share. Our points do not fall with xKV's at high compression, which shows that sharing itself is not the cause of the loss.

Table 1: Protocol B: per-task retention of full-cache accuracy (%) for every operating point of every training-free cross-layer method. Rows are grouped by measured compression tier. Within each tier, the highest compression ratio is always that of BellowsQ, so a baseline with a higher average also stores more memory. **best** and second average per tier are marked. Task codes and settings: Appendix C.1.

| Method | Comp. | Qsp | MFQ | HQA | MSQ | NQA | Gov | QMS | PsR | LCC | RBP | 2Wk | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Llama-3.1-8B-Instruct** | | | | | | | | | | | | | |
| MiniCache fp16 | 1.31× | 72.1 | 72.3 | 74.7 | 86.8 | 55.0 | 82.7 | 89.5 | 70.7 | 80.4 | 82.6 | 55.6 | 74.8 |
| KVSharer fp16 | 1.33× | 91.5 | 96.9 | 93.9 | 98.1 | 78.2 | 73.7 | 97.9 | 99.5 | 85.4 | 84.5 | 70.2 | 88.2 |
| **BellowsQ $n$=2 INT8** | 3.64× | 95.6 | 97.6 | 99.6 | 99.4 | 95.9 | 98.5 | 102.5 | 100.0 | 98.7 | 99.0 | 101.0 | **98.9** |
| CommonKV | 3.95× | 92.2 | 90.6 | 98.0 | 100.1 | 89.0 | 89.7 | 96.5 | 99.5 | 98.4 | 97.8 | 91.1 | 94.8 |
| xKV | 4.00× | 96.6 | 98.6 | 96.0 | 102.1 | 97.9 | 97.0 | 100.8 | 100.0 | 99.4 | 99.4 | 92.1 | 98.2 |
| MiniCache+INT4 | 4.78× | 70.2 | 70.9 | 70.4 | 80.2 | 51.8 | 81.1 | 89.9 | 70.8 | 79.8 | 82.7 | 56.4 | 73.1 |
| KVSharer+INT4 | 4.99× | 91.1 | 94.8 | 94.0 | 99.5 | 79.7 | 72.4 | 97.8 | 99.5 | 85.2 | 83.3 | 70.0 | 87.9 |
| XQuant published | 6.59× | 46.8 | 64.5 | 71.0 | 59.1 | 49.6 | 69.3 | 82.0 | 6.0 | 57.2 | 64.5 | 42.4 | 55.7 |
| MiniCache all+INT4 | 6.72× | 13.4 | 13.2 | 9.6 | 13.9 | 7.5 | 20.7 | 50.6 | 30.1 | 51.4 | 58.3 | 4.4 | 24.8 |
| **BellowsQ $n$=2 INT4** | 6.90× | 96.3 | 97.7 | 100.3 | 97.6 | 98.6 | 98.6 | 101.1 | 100.0 | 99.6 | 99.2 | 101.6 | **99.2** |
| CommonKV | 7.89× | 86.0 | 82.6 | 91.1 | 92.7 | 85.4 | 77.8 | 90.9 | 99.2 | 81.2 | 88.5 | 78.1 | 86.7 |
| BellowsQ $n$=3 INT4 | 10.40× | 93.8 | 96.7 | 99.1 | 102.5 | 97.2 | 97.0 | 99.9 | 100.0 | 99.4 | 98.9 | 100.3 | 98.6 |
| KVSharer+INT2 | 11.24× | 53.6 | 67.4 | 63.9 | 79.0 | 69.2 | 30.8 | 78.2 | 57.8 | 59.7 | 59.3 | 37.0 | 59.6 |
| MiniCache+INT2 | 11.96× | 11.1 | 13.1 | 5.4 | 10.4 | 9.4 | 20.0 | 39.6 | 11.3 | 46.4 | 56.0 | 4.0 | 20.6 |
| **BellowsQ $n$=4 INT4** | 13.84× | 94.6 | 99.0 | 99.1 | 96.4 | 96.2 | 93.5 | 98.7 | 100.0 | 99.7 | 99.0 | 101.5 | **98.0** |
| XQuant 1-bit | 15.11× | 0.0 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 12.3 | 15.4 | 0.0 | 2.5 |
| CommonKV | 15.78× | 34.6 | 49.7 | 43.0 | 55.6 | 56.5 | 50.5 | 83.0 | 77.9 | 54.4 | 56.8 | 28.0 | 53.6 |
| xKV | 16.00× | 91.4 | 80.9 | 82.4 | 87.3 | 73.3 | 75.6 | 94.4 | 95.5 | 92.8 | 86.8 | 58.5 | 83.5 |
| **BellowsQ $n$=4 INT3** | 17.65× | 94.2 | 98.6 | 98.0 | 101.3 | 98.8 | 93.6 | 98.0 | 100.0 | 101.1 | 98.9 | 99.5 | **98.4** |
| BellowsQ $n$=4 K3V2 | 20.47× | 91.6 | 96.6 | 97.9 | 100.4 | 99.6 | 93.5 | 98.0 | 100.0 | 99.9 | 99.3 | 96.6 | 97.6 |
| **Mistral-7B-Instruct-v0.3** | | | | | | | | | | | | | |
| MiniCache fp16 | 1.31× | 76.7 | 69.4 | 72.7 | 56.2 | 81.4 | 68.7 | 83.3 | 37.2 | 76.3 | 74.6 | 67.4 | 69.4 |
| KVSharer fp16 | 1.33× | 103.8 | 95.6 | 94.2 | 90.6 | 100.3 | 91.8 | 93.3 | 68.4 | 89.1 | 87.7 | 87.4 | 91.1 |
| **BellowsQ $n$=2 INT8** | 3.64× | 98.2 | 100.6 | 98.9 | 102.1 | 95.9 | 99.4 | 99.7 | 99.5 | 100.4 | 101.0 | 100.3 | **99.6** |
| CommonKV | 3.95× | 99.4 | 101.2 | 96.4 | 89.7 | 92.3 | 89.7 | 95.8 | 100.0 | 97.1 | 100.0 | 100.0 | 96.5 |
| **xKV** | 4.00× | 102.0 | 100.0 | 100.2 | 100.4 | 95.2 | 98.2 | 99.5 | 95.9 | 98.3 | 100.2 | 103.2 | **99.4** |
| MiniCache+INT4 | 4.78× | 79.0 | 68.6 | 70.6 | 52.7 | 81.4 | 67.9 | 84.0 | 38.8 | 76.5 | 74.8 | 65.4 | 69.1 |
| KVSharer+INT4 | 4.99× | 101.1 | 97.1 | 95.1 | 90.2 | 92.9 | 92.5 | 93.2 | 67.9 | 87.6 | 88.0 | 89.0 | 90.4 |
| XQuant published | 6.59× | 59.1 | 67.7 | 72.8 | 51.1 | 69.9 | 74.9 | 86.9 | 13.3 | 68.2 | 70.4 | 73.5 | 64.3 |
| MiniCache all+INT4 | 6.72× | 35.2 | 23.3 | 22.1 | 23.4 | 39.1 | 18.9 | 50.7 | 19.0 | 49.5 | 42.8 | 30.4 | 32.2 |
| BellowsQ $n$=2 INT4 | 7.02× | 98.4 | 98.8 | 98.3 | 101.5 | 95.6 | 98.3 | 99.4 | 98.0 | 99.8 | 100.8 | 101.6 | 99.1 |
| CommonKV | 7.90× | 90.9 | 93.4 | 98.1 | 84.1 | 101.5 | 79.1 | 95.7 | 95.9 | 93.7 | 93.4 | 94.6 | 92.8 |
| BellowsQ $n$=3 INT4 | 10.46× | 96.6 | 100.8 | 99.7 | 100.5 | 92.1 | 94.9 | 96.7 | 96.9 | 98.5 | 99.9 | 99.5 | 97.8 |
| KVSharer 37.5%+INT2 | 11.24× | 90.3 | 79.8 | 82.4 | 73.5 | 89.9 | 78.3 | 88.1 | 17.3 | 77.3 | 69.4 | 70.1 | 74.2 |
| MiniCache+INT2 | 11.99× | 34.6 | 18.8 | 21.2 | 23.3 | 28.8 | 21.4 | 45.4 | 11.2 | 43.2 | 40.9 | 23.0 | 28.3 |
| **BellowsQ $n$=4 INT4** | 13.84× | 93.9 | 103.1 | 99.1 | 97.2 | 96.3 | 93.6 | 99.8 | 97.4 | 98.6 | 101.0 | 99.3 | **98.1** |
| KVSharer+INT2 | 14.05× | 59.3 | 67.9 | 74.6 | 56.6 | 66.2 | 46.5 | 74.6 | 13.9 | 54.9 | 43.2 | 61.7 | 56.3 |
| XQuant 1-bit | 15.11× | 0.0 | 0.5 | 0.0 | 0.0 | 0.0 | 0.1 | 0.0 | 0.0 | 17.2 | 13.0 | 0.0 | 2.8 |
| CommonKV | 15.80× | 53.1 | 55.8 | 24.9 | 21.1 | 48.8 | 60.2 | 81.0 | 68.9 | 66.2 | 64.3 | 32.0 | 52.4 |
| xKV | 16.00× | 90.4 | 88.5 | 47.8 | 42.1 | 85.2 | 93.0 | 91.8 | 81.7 | 92.7 | 90.0 | 71.1 | 79.5 |
| BellowsQ $n$=4 INT3 | 17.65× | 94.8 | 101.0 | 99.7 | 95.9 | 96.0 | 93.8 | 97.7 | 96.4 | 97.9 | 101.7 | 100.5 | 97.8 |
| **BellowsQ $n$=4 K3V2** | 20.47× | 96.8 | 100.5 | 99.6 | 95.0 | 97.0 | 93.7 | 99.5 | 94.4 | 99.0 | 101.9 | 99.1 | **97.9** |
| **Qwen2.5-7B-Instruct** | | | | | | | | | | | | | |
| MiniCache fp16 | 1.31× | 25.2 | 28.8 | 19.9 | 18.9 | 25.1 | 55.7 | 66.1 | 3.2 | 42.5 | 44.2 | 15.8 | 31.4 |
| KVSharer fp16 | 1.33× | 51.5 | 61.5 | 36.5 | 31.5 | 25.4 | 38.9 | 68.3 | 82.8 | 67.0 | 57.5 | 26.1 | 49.7 |
| **BellowsQ $n$=2 INT8** | 3.53× | 101.0 | 99.1 | 100.4 | 98.1 | 97.0 | 97.9 | 99.8 | 100.0 | 96.6 | 97.1 | 100.0 | **98.8** |
| CommonKV | 3.95× | 93.0 | 89.6 | 94.6 | 86.9 | 99.9 | 86.2 | 92.6 | 98.5 | 89.8 | 95.0 | 88.9 | 92.3 |
| **xKV**† | 4.00× | 97.9 | 101.9 | 96.9 | – | 109.8 | 97.2 | 102.1 | 100.0 | 92.3 | 96.1 | 96.8 | **99.1** |
| MiniCache+INT4 | 4.78× | 24.8 | 30.1 | 20.4 | 17.2 | 24.8 | 56.5 | 64.3 | 4.0 | 44.2 | 43.3 | 13.8 | 31.2 |
| KVSharer+INT4 | 4.99× | 60.2 | 60.7 | 36.1 | 33.3 | 30.4 | 41.4 | 67.5 | 82.1 | 63.5 | 56.3 | 22.9 | 50.4 |
| XQuant published | 6.59× | 34.0 | 39.8 | 41.4 | 27.5 | 29.3 | 50.0 | 75.1 | 4.1 | 30.9 | 35.4 | 28.8 | 36.0 |
| BellowsQ $n$=2 INT4 | 6.66× | 101.6 | 100.2 | 99.2 | 97.3 | 97.2 | 97.3 | 100.5 | 100.0 | 97.7 | 98.1 | 98.6 | 98.9 |
| MiniCache all+INT4 | 6.72× | 14.1 | 14.4 | 10.8 | 8.7 | 17.3 | 34.6 | 63.0 | 0.2 | 36.2 | 35.5 | 9.8 | 22.2 |
| CommonKV | 7.89× | 59.0 | 59.0 | 60.8 | 49.9 | 73.7 | 66.3 | 88.6 | 72.5 | 69.5 | 70.5 | 35.0 | 64.1 |
| xKV† | 8.00× | 94.3 | 96.6 | 85.7 | 78.9 | 73.8 | 93.7 | 99.4 | 98.5 | 91.9 | – | 85.0 | 89.8 |
| BellowsQ $n$=3 INT4 | 9.92× | 100.0 | 96.5 | 99.5 | 98.4 | 101.6 | 94.3 | 100.5 | 99.5 | 95.3 | 96.1 | 100.7 | 98.4 |
| KVSharer+INT2 | 10.93× | 19.0 | 30.9 | 25.3 | 16.6 | 25.9 | 32.0 | 50.8 | 15.0 | 36.0 | 36.9 | 9.9 | 27.1 |
| MiniCache+INT2 | 11.97× | 13.0 | 12.6 | 8.3 | 7.6 | 17.9 | 39.9 | 56.9 | 1.4 | 35.3 | 30.9 | 7.4 | 21.0 |
| **BellowsQ $n$=4 INT4** | 13.13× | 100.8 | 97.3 | 98.6 | 99.2 | 96.8 | 93.6 | 99.9 | 99.0 | 92.6 | 94.5 | 98.6 | **97.3** |
| XQuant 1-bit | 15.11× | 0.7 | 2.1 | 0.1 | 0.6 | 1.3 | 9.8 | 12.7 | 1.0 | 21.2 | 24.1 | 0.1 | 6.7 |
| CommonKV | 15.78× | 15.2 | 22.9 | 31.1 | 22.6 | 22.9 | 47.7 | 66.4 | 5.5 | 38.6 | 42.8 | 11.7 | 29.8 |
| xKV | 16.00× | 63.0 | 69.6 | 26.1 | 22.2 | 38.2 | 80.8 | 78.3 | 48.5 | 74.7 | 61.5 | 35.9 | 54.4 |
| **BellowsQ $n$=4 INT3** | 16.52× | 98.2 | 98.8 | 98.6 | 96.7 | 97.2 | 93.3 | 101.4 | 98.0 | 94.5 | 93.3 | 97.1 | **97.0** |
| BellowsQ $n$=4 K3V2 | 18.96× | 99.6 | 95.8 | 99.6 | 98.8 | 100.0 | 93.2 | 100.0 | 98.0 | 88.4 | 94.6 | 96.3 | 96.8 |

**Beyond LongBench.** Protocol C repeats the comparison on an agent task that must read its retrieved evidence verbatim, on two models the paper has not used, one of them a mixture of experts (Table 2), and there the rank cliff arrives two octaves earlier: xKV scores 26.5 at 8× and, on the mixture of experts, 15.5 at 4×, and CommonKV has lost half of dense by 8×. Cutting bits at full rank still costs nothing, with BellowsQ at 86.0 and 85.8 at 14.6× against dense 85.5 and 86.1, so the rule holds where the task is least forgiving.

Table 2: Protocol C: single-turn retrieve-then-read on BrowseComp-Plus on all 830 queries, with the oracle prompt at a 32,768-token budget. Results are scored by the benchmark's own judge. The notation "–" means that the method cannot reach that compression rate. See full table in Appendix C.3

| | | | cross-layer latent | | | per-layer quantized | | |
|---|---|---|---|---|---|---|---|---|
| Model | Tier | Dense | **BellowsQ** | CommonKV | xKV | OTT | KIVI | PolarQuant |
| Qwen3-32B | ∼8× | 85.54 | **84.22** | 81.08 | 26.51 | 77.95 | 75.42 | 51.93 |
| | ∼16× | 85.54 | **86.02** | 66.02 | 8.43 | – | – | – |
| Qwen3-30B-A3B | ∼8× | 86.14 | **85.30** | 49.88 | 5.30 | 81.45 | 77.59 | 4.22 |
| | ∼16× | 86.14 | **85.78** | – | – | – | – | – |

Table 3: Cutting rank is the cliff; the rival that holds its rank and cuts bits instead comes back. (See full table in Appendix 19)

| Arm | Comp. | Llama | Mistral | Qwen |
|---|---|---|---|---|
| xKV, fp16, rank $p$ | $4.0\times$ | 98.2 | 99.4 | 99.1 |
| xKV, fp16, rank $p/2$ | $8.0\times$ | – | – | 89.8 |
| xKV, fp16, rank $p/4$ | $16.0\times$ | 83.5 | 79.5 | 54.4 |
| **xKV-hold**: the same basis, near-full rank, our INT4 leg | $15.1\times$ | 97.6 | 99.2 | 23.4 |
| BellowsQ $n$=4 INT4 (the frozen consensus) | $13.1$–$13.8\times$ | 98.0 | 98.1 | 97.3 |
| our construction, per-sequence SVD oracle ($2.2\times$ prefill) | $15.1\times$ | 97.3 | – | – |
| CommonKV, rank 81% of the width | $4.0\times$ | 94.8 | 96.5 | 92.3 |
| CommonKV, rank 20% of the width | $15.8\times$ | 53.6 | 52.4 | 29.8 |

### 4.2 Dimension and bit-width at a fixed budget

We compare two ways to reach a high compression ratio at a fixed memory budget: reducing the dimension of the shared basis, and reducing the number of bits per stored value. We run this comparison on the construction of xKV, because both settings exist for it (Table 3). xKV reaches $16\times$ by reducing its per-sequence basis from rank $p$ to $p/4$. xKV-hold keeps the same basis at near-full rank and stores it at INT4, which gives $15.1\times$. A third setting replaces our fixed projection with a per-sequence SVD, at $2.2\times$ the prefill cost.

xKV-hold reaches $97.6$ on Llama and $99.2$ on Mistral, against $83.5$ and $79.5$ for xKV at $16\times$. The per-sequence SVD reaches $97.3$ on Llama, which is below the $98.0$ of the fixed projection. On Qwen, every setting that stores xKV's basis at four bits fails, and we have no explanation for this case. The results indicate that the loss of xKV at $16\times$ is caused by the reduced dimension. xKV-hold keeps the same shared basis, so the sharing itself is not the cause.

### 4.3 Comparison with per-layer quantizers

Protocol A compares the shared latent with per-layer quantizers at matched memory. We run every official configuration of KIVI, OTT, RotateKV, TurboQuant and PolarQuant (Liu et al., 2024b; Su et al., 2025a;b; Zandieh et al., 2025; Han et al., 2025), and every BellowsKV point from $1.92\times$ to $18.6\times$ (Table 4).

On the latent at $n$=4, the loss from fp16 to INT4 is $0.1$ to $0.3$ points, and the loss from INT4 to INT3 is at most $1.1$ points. For the rotated per-layer caches, the loss from four bits to two bits is $4.1$ to $6.4$ points. Note that these two comparisons do not use the same change in bit-width; the comparison at equal bit-width is given in Table 18. At $\sim 7\times$ on Llama, OTT at two bits reaches $53.30$ against $52.92$ for BellowsKV, while storing $5\%$ fewer bytes. Above $8\times$, no per-layer configuration exists, because two bits at full dimension already gives $8\times$.

We also compare one eviction decision per block with the same rule applied independently in each layer (Table 5). Both routes use the same scoring rule. The per-layer baseline scores each layer on its own fp16 cache. The full grid is given in Appendix C.4.

At the same keep ratio, the per-layer baseline is better by $0.8$ to $3.3$ points, and it stores $3.8\times$ more memory. At equal memory, the block-level decision is better by $0.9$ to $3.0$ points at about $8\times$, and by $5.3$ to $6.6$ points at about $16\times$. The advantage is larger at the higher compression ratio.

Table 4: Protocol A: LongBench average over eight tasks under the official harness. The table reports every official configuration of KIVI, OTT, RotateKV, TurboQuant and PolarQuant, and every BellowsKV point, grouped by measured compression tier. Within every tier, the highest compression ratio is that of BellowsKV, so a per-layer quantizer with a higher average also stores more memory. No per-layer point exists above $8\times$, because two bits at full dimension already give $8\times$. Per-task results: Appendix C.2. Equal bit-width comparison: Table 18.

| Method | Comp. | Llama | Mistral | Qwen |
|---|---|---|---|---|
| Dense fp16 | 1.00× | 53.29 | 52.78 | 53.23 |
| BellowsS $n$=2 | 1.9× | 53.07 | **53.04** | 51.88 |
| BellowsS $n$=3 | 2.9–2.9× | 52.67 | 52.87 | 51.92 |
| BellowsQ $n$=2 INT8 | 3.6–3.7× | 53.05 | 52.92 | 52.00 |
| PolarQuant r4t4v4 | 3.7× | 53.66 | – | **53.90** |
| KIVI K4V4 | 3.8× | **53.92** | 52.80 | – |
| OTT K4V4 | 3.8× | 53.83 | 52.99 | – |
| BellowsS $n$=4 | 3.8–3.9× | 52.78 | 53.04 | 51.21 |
| TurboQuant K4V4 | 4.0× | 52.84 | 52.44 | 52.98 |
| RotateKV 4-bit | 4.0× | – | **53.14** | – |
| PolarQuant r3t3v4 | 4.1× | **53.50** | – | **53.89** |
| PolarQuant r2t2v4 | 4.7× | 50.63 | – | 50.87 |
| RotateKV 3-bit | 5.3× | 52.38 | 52.33 | – |
| TurboQuant K4V2 | 5.3× | 52.57 | 51.84 | 51.25 |
| BellowsQ $n$=3 INT8 | 5.3–5.5× | 52.84 | 53.04 | 51.90 |
| BellowsQ $n$=2 INT4 | 6.4–6.8× | 52.92 | **52.76** | **51.86** |
| PolarQuant r2t2v2 | 6.7× | 48.51 | – | 50.12 |
| KIVI K2V2 | 7.1× | 52.17 | 52.13 | – |
| OTT K2V2 | 7.1× | **53.30** | 52.51 | – |
| TurboQuant K2V2 | 8.0× | 46.43 | 48.31 | 47.18 |
| RotateKV 2-bit | 8.0× | 47.00 | 49.01 | – |
| BellowsQ $n$=2 INT3 | 8.0–8.6× | 52.72 | 52.67 | 51.60 |
| BellowsQ $n$=4 INT4 | 12.2–13.0× | **52.45** | 52.83 | **51.12** |
| BellowsQ $n$=4 INT3 | 15.0–16.2× | 52.42 | 52.75 | 50.02 |
| BellowsQ $n$=4 K3V2 | 16.9–18.6× | 52.27 | **52.87** | 50.28 |

Table 5: Protocol B retention (%) of full-cache accuracy. All routes use the same scoring rule. The per-layer baseline applies this rule independently in each layer, on the fp16 cache of that layer. The comparison at equal keep ratio is given in Table 15.

| Budget | Route | Keep | Comp. | Llama | Mistral | Qwen |
|---|---|---|---|---|---|---|
| ∼8× | BellowsE | 1/2 | 7.7–7.8 | **97.6** | **97.9** | **99.2** |
| | Per-layer eviction | 1/8 | 8.0 | 94.7 | 94.9 | 98.3 |
| ∼16× | BellowsE | 1/4 | 15.2–15.4 | **96.1** | **96.4** | **97.9** |
| | Per-layer eviction | 1/16 | 16.0 | 89.5 | 90.7 | 92.6 |
| ∼30× | BellowsQ×BellowsE | 1/2 | 26.3–27.7 | **97.5** | 97.9 | **99.2** |
| | BellowsE | 1/8 | 29.6–30.0 | 92.5 | 93.8 | 95.0 |
| | xKV+INT4+pooled SnapKV | 1/2 | 30.1 | 96.1 | **98.4** | 23.8 |

### 4.4 Decode cost

The introduction promised a decode cost, with a relation bounding any reconstruct-at-read design: decode speedup $= (d_{kv}/R) \times 2/(\text{bytes per latent value at read})$. $n$ is absent, so sharing buys capacity, never speed, and a design holding $R{=}d_{kv}$ that reads fp16 should decode at parity. On an H100 (HuggingFace, 32k context, no custom kernel) BellowsKV decodes at $0.995\times$ dense against a predicted $1.00\times$; Palu (Chang et al., 2024), the rank-cutting design with a published decode figure, reports $1.89\times$ against a predicted $2.00\times$ (Appendix Table 6). The freed bytes buy a $2\times$ batch under HuggingFace and a $14.8\times$ KV pool under vLLM (above the $13.8\times$ byte ratio for a reason we have not isolated). The measured decode speed is $0.995\times$ that of the uncompressed cache. This confirms two properties of the design: reconstruction at read adds no measurable cost, and sharing across layers does not change decode speed.

## 5 Conclusion

BellowsKV starts from one measured property: what a block of layers shares is the stream behind its caches, and the aggregate drawn from it is a consensus, tame in amplitude, reliable in judgment, irreducible in dimension. One training-free construction turns it into a compressor whose two operators compose: near-dense accuracy at ratios where published cross-layer methods fail, at the decode speed of an uncompressed cache. The takeaway is the rule the experiments price: hold the rank, buy bits first, evict last.

### AI USE STATEMENT

We used generative AI tools, in the form of an interactive coding assistant, help with implementing methods (checking for bugs, environment installation and package version checks). We also use it to help check and revise grammar mistake, We did not use generative AI tools to generate synthetic data, to develop the conceptual framework or the method itself, or assist in proving them, or to propose or refine hypotheses.

### REPRODUCIBILITY STATEMENT

The construction of the shared latent and the two operators is given in Section 3, with the full derivation in Appendix B. The calibration procedure, the evaluation protocols, and the configuration of every baseline are given in Appendix C.

## REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints, 2023. URL https://arxiv.org/abs/2305.13245.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms, 2024. URL https://arxiv.org/abs/2404.00456.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding, 2024. URL https://arxiv.org/abs/2308.14508.

William Brandon, Mayank Mishra, Aniruddha Nrusimha, Rameswar Panda, and Jonathan Ragan Kelly. Reducing transformer key-value cache size with cross-layer attention, 2024. URL https://arxiv.org/abs/2405.12981.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed S. Abdelfattah, and Kai-Chiang Wu. Palu: Compressing kv-cache with low-rank projection, 2024. URL https://arxiv.org/abs/2407.21118.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Hung-Yueh Chiang, Yash Akhauri, Xilai Dai, Huiqiang Jiang, Yucheng Li, Luis Ceze, Kai-Chiang Wu, and Mohamed S. Abdelfattah. xkv: Cross-layer kv-cache compression via aligned singular vector extraction, 2026. URL https://arxiv.org/abs/2503.18893.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Sahel Sharifymoghaddam, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent, 2025. URL https://arxiv.org/abs/2508.06600.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jin Chen, Jingyang Yuan, Junjie Qiu, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruizhe Pan, Runxin Xu, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Size Zheng, T. Wang, Tian Pei, Tian Yuan, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang

Chen, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Liu, Xin Xie, Xingkai Yu, Xinnan Song, Xinyi Zhou, Xinyu Yang, Xuan Lu, Xuecheng Su, Y. Wu, Y. K. Li, Y. X. Wei, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Zheng, Yichao Zhang, Yiliang Xiong, Yilong Zhao, Ying He, Ying Tang, Yishi Piao, Yixin Dong, Yixuan Tan, Yiyuan Liu, Yongji Wang, Yongqiang Guo, Yuchen Zhu, Yuduan Wang, Yuheng Zou, Yukun Zha, Yunxian Ma, Yuting Yan, Yuxiang You, Yuxuan Liu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhewen Hao, Zhihong Shao, Zhiniu Wen, Zhipeng Xu, Zhongyu Zhang, Zhuoshu Li, Zihan Wang, Zihui Gu, Zilin Li, and Ziwei Xie. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024. URL https://arxiv.org/abs/2405.04434.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

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, Jennifer 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, 2024. URL https://arxiv.org/abs/2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu,

Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645 (8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, 2010. URL https://arxiv.org/abs/0909.4061.

Insu Han, Praneeth Kacham, Amin Karbasi, Vahab Mirrokni, and Amir Zandieh. Polarquant: Quantizing kv caches with polar transformation, 2025. URL https://arxiv.org/abs/2502.02617.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models?, 2024. URL https://arxiv.org/abs/2404.06654.

Alan Julian Izenman. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248–264, 1975.

Tao Ji, Bin Guo, Yuanbin Wu, Qipeng Guo, Lixing Shen, Zhan Chen, Xipeng Qiu, Qi Zhang, and Tao Gui. Towards economical inference: Enabling deepseek's multi-head latent attention in any transformer-based llms, 2025. URL https://arxiv.org/abs/2502.14837.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues?, 2024. URL https://arxiv.org/abs/2310.06770.

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, 2024. URL https://arxiv.org/abs/2404.14469.

Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. Minicache: Kv cache compression in depth dimension for large language models, 2024a. URL https://arxiv.org/abs/2405.14366.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations, 2025. URL https://arxiv.org/abs/2405.16406.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *International Conference on Machine Learning (ICML)*, 2024b.

Da Ma, Lu Chen, Situo Zhang, Yuxun Miao, Su Zhu, Zhi Chen, Hongshen Xu, Hanqi Li, Shuai Fan, Lei Pan, and Kai Yu. Compressing kv cache for long-context llm inference with inter-layer attention similarity, 2025. URL https://arxiv.org/abs/2412.02252.

Fanxu Meng, Zengwei Yao, and Muhan Zhang. Transmla: Multi-head latent attention is all you need, 2025. URL https://arxiv.org/abs/2502.07864.

Leon Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11(1):50–59, 1960.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Bohan Zhang, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wenting Zhan, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2026. URL https://arxiv.org/abs/2412.16720.

Samuel Salfati. Quantization dominates rank reduction for kv-cache compression, 2026. URL https://arxiv.org/abs/2604.11501.

Utkarsh Saxena, Gobinda Saha, Sakshi Choudhary, and Kaushik Roy. Eigen attention: Attention in low-rank space for kv cache compression, 2024. URL https://arxiv.org/abs/2408.05646.

Konrad Staniszewski and Adrian Łańcucki. Kv cache transform coding for compact storage in llm inference, 2026. URL https://arxiv.org/abs/2511.01815.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Yi Su, Yuechi Zhou, Quantong Qiu, Juntao Li, Qingrong Xia, Ping Li, Xinyu Duan, Zhefeng Wang, and Min Zhang. Accurate kv cache quantization with outlier tokens tracing, 2025a. URL https://arxiv.org/abs/2505.10938.

Zunhai Su, Zhe Chen, Wang Shen, Hanyu Wei, Linge Li, Huangqi Yu, and Kehong Yuan. Rotatekv: Accurate and robust 2-bit kv cache quantization for llms via outlier-aware adaptive rotations, 2025b. URL https://arxiv.org/abs/2501.16383.

Zunhai Su, Rui Yang, Chao Zhang, Yaxiu Liu, Yifan Zhang, Wei Wu, Jing Xiong, Dayou Du, Xialie Zhuang, Yulei Qian, Yuchen Xie, Yik-Chung Wu, Hongxia Yang, and Ngai Wong. Oscar: The occam's razor for extreme kv cache quantization in llms and beyond, 2026. URL https://arxiv.org/abs/2605.19660.

Yutao Sun, Li Dong, Yi Zhu, Shaohan Huang, Wenhui Wang, Shuming Ma, Quanlu Zhang, Jianyong Wang, and Furu Wei. You only cache once: Decoder-decoder architectures for language models, 2024. URL https://arxiv.org/abs/2405.05254.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, Colin Gaffney, Samira Daruki, Olcan Sercinoglu, Zach Gleicher, Juliette Love, Paul Voigtlaender, Rohan Jain, Gabriela Surita, Kareem Mohamed, Rory Blevins, Junwhan Ahn, Tao Zhu, Kornraphop Kawintiranon, Orhan Firat, Yiming Gu, Yujing Zhang, Matthew Rahtz, Manaal Faruqui, Natalie Clay, Justin Gilmer, JD Co-Reyes, Ivo Penchev, Rui Zhu, Nobuyuki Morioka, Kevin Hui, Krishna Haridasan, Victor Campos, Mahdis Mahdieh, Mandy Guo, Samer Hassan, Kevin Kilgour, Arpi Vezer, Heng-Tze Cheng, Raoul de Liedekerke, Siddharth Goyal, Paul Barham, DJ Strouse, Seb Noury, Jonas Adler, Mukund Sundararajan, Sharad Vikram, Dmitry Lepikhin, Michela Paganini, Xavier Garcia, Fan Yang, Dasha Valter, Maja Trebacz, Kiran Vodrahalli, Chulayuth Asawaroengchai, Roman Ring, Norbert Kalb, Livio Baldini Soares, Siddhartha Brahma, David Steiner, Tianhe Yu, Fabian Mentzer, Antoine He, Lucas Gonzalez, Bibo Xu, Raphael Lopez Kaufman, Laurent El Shafey, Junhyuk Oh, Tom Hennigan, George van den Driessche, Seth Odoom, Mario Lucic, Becca Roelofs, Sid Lall, Amit Marathe, Betty Chan, Santiago Ontanon, Luheng He, Denis Teplyashin, Jonathan Lai, Phil Crone, Bogdan Damoc, Lewis Ho, Sebastian Riedel, Karel Lenc, Chih-Kuan Yeh, Aakanksha Chowdhery, Yang Xu, Mehran Kazemi, Ehsan Amid, Anastasia Petrushkina, Kevin Swersky, Ali Khodaei, Gowoon Chen, Chris Larkin, Mario Pinto, Geng Yan, Adria Puigdomenech Badia, Piyush Patil, Steven Hansen, Dave Orr, Sebastien M. R. Arnold, Jordan Grimstad, Andrew Dai, Sholto Douglas, Rishika Sinha, Vikas Yadav, Xi Chen, Elena Gribovskaya, Jacob Austin, Jeffrey Zhao, Kaushal Patel, Paul Komarek, Sophia Austin, Sebastian Borgeaud, Linda Friso, Abhimanyu Goyal, Ben Caine, Kris Cao, Da-Woon Chung, Matthew Lamm, Gabe Barth-Maron, Thais Kagohara, Kate Olszewska, Mia Chen, Kaushik Shivakumar, Rishabh Agarwal, Harshal Godhia, Ravi Rajwar, Javier Snaider, Xerxes Dotiwalla, Yuan Liu, Aditya Barua, Victor Ungureanu, Yuan Zhang, Bat-Orgil Batsaikhan, Mateo Wirth, James Qin, Ivo Danihelka, Tulsee Doshi, Martin Chadwick, Jilin Chen, Sanil Jain, Quoc Le, Arjun Kar, Madhu Gurumurthy, Cheng Li, Ruoxin Sang, Fangyu Liu, Lampros Lamprou, Rich Munoz, Nathan Lintz, Harsh Mehta, Heidi Howard, Malcolm Reynolds, Lora Aroyo, Quan Wang, Lorenzo Blanco, Albin Cassirer, Jordan Griffith, Dipanjan Das, Stephan Lee, Jakub Sygnowski, Zach Fisher, James Besley, Richard Powell, Zafarali Ahmed, Dominik Paulus, David Reitter, Zalan Borsos, Rishabh Joshi, Aedan Pope, Steven Hand, Vittorio Selo, Vihan Jain, Nikhil Sethi, Megha Goel, Takaki Makino, Rhys May, Zhen Yang, Johan Schalkwyk, Christina Butterfield, Anja Hauth, Alex Goldin, Will Hawkins, Evan Senter, Sergey Brin, Oliver Woodman,

Marvin Ritter, Eric Noland, Minh Giang, Vijay Bolina, Lisa Lee, Tim Blyth, Ian Mackinnon, Machel Reid, Obaid Sarvana, David Silver, Alexander Chen, Lily Wang, Loren Maggiore, Oscar Chang, Nithya Attaluri, Gregory Thornton, Chung-Cheng Chiu, Oskar Bunyan, Nir Levine, Timothy Chung, Evgenii Eltyshev, Xiance Si, Timothy Lillicrap, Demetra Brady, Vaibhav Aggarwal, Boxi Wu, Yuanzhong Xu, Ross McIlroy, Kartikeya Badola, Paramjit Sandhu, Erica Moreira, Wojciech Stokowiec, Ross Hemsley, Dong Li, Alex Tudor, Pranav Shyam, Elahe Rahimtoroghi, Salem Haykal, Pablo Sprechmann, Xiang Zhou, Diana Mincu, Yujia Li, Ravi Addanki, Kalpesh Krishna, Xiao Wu, Alexandre Frechette, Matan Eyal, Allan Dafoe, Dave Lacey, Jay Whang, Thi Avrahami, Ye Zhang, Emanuel Taropa, Hanzhao Lin, Daniel Toyama, Eliza Rutherford, Motoki Sano, HyunJeong Choe, Alex Tomala, Chalence Safranek-Shrader, Nora Kassner, Mantas Pajarskas, Matt Harvey, Sean Sechrist, Meire Fortunato, Christina Lyu, Gamaleldin Elsayed, Chenkai Kuang, James Lottes, Eric Chu, Chao Jia, Chih-Wei Chen, Peter Humphreys, Kate Baumli, Connie Tao, Rajkumar Samuel, Cicero Nogueira dos Santos, Anders Andreassen, Nemanja Rakićević, Dominik Grewe, Aviral Kumar, Stephanie Winkler, Jonathan Caton, Andrew Brock, Sid Dalmia, Hannah Sheahan, Iain Barr, Yingjie Miao, Paul Natsev, Jacob Devlin, Feryal Behbahani, Flavien Prost, Yanhua Sun, Artiom Myaskovsky, Thanumalayan Sankaranarayana Pillai, Dan Hurt, Angeliki Lazaridou, Xi Xiong, Ce Zheng, Fabio Pardo, Xiaowei Li, Dan Horgan, Joe Stanton, Moran Ambar, Fei Xia, Alejandro Lince, Mingqiu Wang, Basil Mustafa, Albert Webson, Hyo Lee, Rohan Anil, Martin Wicke, Timothy Dozat, Abhishek Sinha, Enrique Piqueras, Elahe Dabir, Shyam Upadhyay, Anudhyan Boral, Lisa Anne Hendricks, Corey Fry, Josip Djolonga, Yi Su, Jake Walker, Jane Labanowski, Ronny Huang, Vedant Misra, Jeremy Chen, RJ Skerry-Ryan, Avi Singh, Shruti Rijhwani, Dian Yu, Alex Castro-Ros, Beer Changpinyo, Romina Datta, Sumit Bagri, Arnar Mar Hrafnkelsson, Marcello Maggioni, Daniel Zheng, Yury Sulsky, Shaobo Hou, Tom Le Paine, Antoine Yang, Jason Riesa, Dominika Rogozinska, Dror Marcus, Dalia El Badawy, Qiao Zhang, Luyu Wang, Helen Miller, Jeremy Greer, Lars Lowe Sjos, Azade Nova, Heiga Zen, Rahma Chaabouni, Mihaela Rosca, Jiepu Jiang, Charlie Chen, Ruibo Liu, Tara Sainath, Maxim Krikun, Alex Polozov, Jean-Baptiste Lespiau, Josh Newlan, Zeyncep Cankara, Soo Kwak, Yunhan Xu, Phil Chen, Andy Coenen, Clemens Meyer, Katerina Tsihlas, Ada Ma, Juraj Gottweis, Jinwei Xing, Chenjie Gu, Jin Miao, Christian Frank, Zeynep Cankara, Sanjay Ganapathy, Ishita Dasgupta, Steph Hughes-Fitt, Heng Chen, David Reid, Keran Rong, Hongmin Fan, Joost van Amersfoort, Vincent Zhuang, Aaron Cohen, Shixiang Shane Gu, Anhad Mohananey, Anastasija Ilic, Taylor Tobin, John Wieting, Anna Bortsova, Phoebe Thacker, Emma Wang, Emily Caveness, Justin Chiu, Eren Sezener, Alex Kaskasoli, Steven Baker, Katie Millican, Mohamed Elhawaty, Kostas Aisopos, Carl Lebsack, Nathan Byrd, Hanjun Dai, Wenhao Jia, Matthew Wiethoff, Elnaz Davoodi, Albert Weston, Lakshman Yagati, Arun Ahuja, Isabel Gao, Golan Pundak, Susan Zhang, Michael Azzam, Khe Chai Sim, Sergi Caelles, James Keeling, Abhanshu Sharma, Andy Swing, YaGuang Li, Chenxi Liu, Carrie Grimes Bostock, Yamini Bansal, Zachary Nado, Ankesh Anand, Josh Lipschultz, Abhijit Karmarkar, Lev Proleev, Abe Ittycheriah, Soheil Hassas Yeganeh, George Polovets, Aleksandra Faust, Jiao Sun, Alban Rrustemi, Pen Li, Rakesh Shivanna, Jeremiah Liu, Chris Welty, Federico Lebron, Anirudh Baddepudi, Sebastian Krause, Emilio Parisotto, Radu Soricut, Zheng Xu, Dawn Bloxwich, Melvin Johnson, Behnam Neyshabur, Justin Mao-Jones, Renshen Wang, Vinay Ramasesh, Zaheer Abbas, Arthur Guez, Constant Segal, Duc Dung Nguyen, James Svensson, Le Hou, Sarah York, Kieran Milan, Sophie Bridgers, Wiktor Gworek, Marco Tagliasacchi, James Lee-Thorp, Michael Chang, Alexey Guseynov, Ale Jakse Hartman, Michael Kwong, Ruizhe Zhao, Sheleem Kashem, Elizabeth Cole, Antoine Miech, Richard Tanburn, Mary Phuong, Filip Pavetic, Sebastien Cevey, Ramona Comanescu, Richard Ives, Sherry Yang, Cosmo Du, Bo Li, Zizhao Zhang, Mariko Iinuma, Clara Huiyi Hu, Aurko Roy, Shaan Bijwadia, Zhenkai Zhu, Danilo Martins, Rachel Saputro, Anita Gergely, Steven Zheng, Dawei Jia, Ioannis Antonoglou, Adam Sadovsky, Shane Gu, Yingying Bi, Alek Andreev, Sina Samangooei, Mina Khan, Tomas Kocisky, Angelos Filos, Chintu Kumar, Colton Bishop, Adams Yu, Sarah Hodkinson, Sid Mittal, Premal Shah, Alexandre Moufarek, Yong Cheng, Adam Bloniarz, Jaehoon Lee, Pedram Pejman, Paul Michel, Stephen Spencer, Vladimir Feinberg, Xuehan Xiong, Nikolay Savinov, Charlotte Smith, Siamak Shakeri, Dustin Tran, Mary Chesus, Bernd Bohnet, George Tucker, Tamara von Glehn, Carrie Muir, Yiran Mao, Hideto Kazawa, Ambrose Slone, Kedar Soparkar, Disha Shrivastava, James Cobon-Kerr, Michael Sharman, Jay Pavagadhi, Carlos Araya, Karolis Misiunas, Nimesh Ghelani, Michael Laskin, David Barker, Qiujia Li, Anton Briukhov, Neil Houlsby, Mia Glaese, Balaji Lakshminarayanan, Nathan Schucher, Yunhao Tang, Eli Collins, Hyeontaek Lim, Fangxiaoyu Feng, Adria Recasens, Guangda Lai, Alberto Magni, Nicola De Cao, Aditya Siddhant, Zoe Ashwood,

Jordi Orbay, Mostafa Dehghani, Jenny Brennan, Yifan He, Kelvin Xu, Yang Gao, Carl Saroufim, James Molloy, Xinyi Wu, Seb Arnold, Solomon Chang, Julian Schrittwieser, Elena Buchatskaya, Soroush Radpour, Martin Polacek, Skye Giordano, Ankur Bapna, Simon Tokumine, Vincent Hellendoorn, Thibault Sottiaux, Sarah Cogan, Aliaksei Severyn, Mohammad Saleh, Shantanu Thakoor, Laurent Shefey, Siyuan Qiao, Meenu Gaba, Shuo yiin Chang, Craig Swanson, Biao Zhang, Benjamin Lee, Paul Kishan Rubenstein, Gan Song, Tom Kwiatkowski, Anna Koop, Ajay Kannan, David Kao, Parker Schuh, Axel Stjerngren, Golnaz Ghiasi, Gena Gibson, Luke Vilnis, Ye Yuan, Felipe Tiengo Ferreira, Aishwarya Kamath, Ted Klimenko, Ken Franko, Kefan Xiao, Indro Bhattacharya, Miteyan Patel, Rui Wang, Alex Morris, Robin Strudel, Vivek Sharma, Peter Choy, Sayed Hadi Hashemi, Jessica Landon, Mara Finkelstein, Priya Jhakra, Justin Frye, Megan Barnes, Matthew Mauger, Dennis Daun, Khuslen Baatarsukh, Matthew Tung, Wael Farhan, Henryk Michalewski, Fabio Viola, Felix de Chaumont Quitry, Charline Le Lan, Tom Hudson, Qingze Wang, Felix Fischer, Ivy Zheng, Elspeth White, Anca Dragan, Jean baptiste Alayrac, Eric Ni, Alexander Pritzel, Adam Iwanicki, Michael Isard, Anna Bulanova, Lukas Zilka, Ethan Dyer, Devendra Sachan, Srivatsan Srinivasan, Hannah Muckenhirn, Honglong Cai, Amol Mandhane, Mukarram Tariq, Jack W. Rae, Gary Wang, Kareem Ayoub, Nicholas FitzGerald, Yao Zhao, Woohyun Han, Chris Alberti, Dan Garrette, Kashyap Krishnakumar, Mai Gimenez, Anselm Levskaya, Daniel Sohn, Josip Matak, Inaki Iturrate, Michael B. Chang, Jackie Xiang, Yuan Cao, Nishant Ranka, Geoff Brown, Adrian Hutter, Vahab Mirrokni, Nanxin Chen, Kaisheng Yao, Zoltan Egyed, Francois Galilee, Tyler Liechty, Praveen Kallakuri, Evan Palmer, Sanjay Ghemawat, Jasmine Liu, David Tao, Chloe Thornton, Tim Green, Mimi Jasarevic, Sharon Lin, Victor Cotruta, Yi-Xuan Tan, Noah Fiedel, Hongkun Yu, Ed Chi, Alexander Neitz, Jens Heitkaemper, Anu Sinha, Denny Zhou, Yi Sun, Charbel Kaed, Brice Hulse, Swaroop Mishra, Maria Georgaki, Sneha Kudugunta, Clement Farabet, Izhak Shafran, Daniel Vlasic, Anton Tsitsulin, Rajagopal Ananthanarayanan, Alen Carin, Guolong Su, Pei Sun, Shashank V, Gabriel Carvajal, Josef Broder, Iulia Comsa, Alena Repina, William Wong, Warren Weilun Chen, Peter Hawkins, Egor Filonov, Lucia Loher, Christoph Hirnschall, Weiyi Wang, Jingchen Ye, Andrea Burns, Hardie Cate, Diana Gage Wright, Federico Piccinini, Lei Zhang, Chu-Cheng Lin, Ionel Gog, Yana Kulizhskaya, Ashwin Sreevatsa, Shuang Song, Luis C. Cobo, Anand Iyer, Chetan Tekur, Guillermo Garrido, Zhuyun Xiao, Rupert Kemp, Huaixiu Steven Zheng, Hui Li, Ananth Agarwal, Christel Ngani, Kati Goshvadi, Rebeca Santamaria-Fernandez, Wojciech Fica, Xinyun Chen, Chris Gorgolewski, Sean Sun, Roopal Garg, Xinyu Ye, S. M. Ali Eslami, Nan Hua, Jon Simon, Pratik Joshi, Yelin Kim, Ian Tenney, Sahitya Potluri, Lam Nguyen Thiet, Quan Yuan, Florian Luisier, Alexandra Chronopoulou, Salvatore Scellato, Praveen Srinivasan, Minmin Chen, Vinod Koverkathu, Valentin Dalibard, Yaming Xu, Brennan Saeta, Keith Anderson, Thibault Sellam, Nick Fernando, Fantine Huot, Junehyuk Jung, Mani Varadarajan, Michael Quinn, Amit Raul, Maigo Le, Ruslan Habalov, Jon Clark, Komal Jalan, Kalesha Bullard, Achintya Singhal, Thang Luong, Boyu Wang, Sujeevan Rajayogam, Julian Eisenschlos, Johnson Jia, Daniel Finchelstein, Alex Yakubovich, Daniel Balle, Michael Fink, Sameer Agarwal, Jing Li, Dj Dvijotham, Shalini Pal, Kai Kang, Jaclyn Konzelmann, Jennifer Beattie, Olivier Dousse, Diane Wu, Remi Crocker, Chen Elkind, Siddhartha Reddy Jonnalagadda, Jong Lee, Dan Holtmann-Rice, Krystal Kallarackal, Rosanne Liu, Denis Vnukov, Neera Vats, Luca Invernizzi, Mohsen Jafari, Huanjie Zhou, Lilly Taylor, Jennifer Prendki, Marcus Wu, Tom Eccles, Tianqi Liu, Kavya Kopparapu, Francoise Beaufays, Christof Angermueller, Andreea Marzoca, Shourya Sarcar, Hilal Dib, Jeff Stanway, Frank Perbet, Nejc Trdin, Rachel Sterneck, Andrey Khorlin, Dinghua Li, Xihui Wu, Sonam Goenka, David Madras, Sasha Goldshtein, Willi Gierke, Tong Zhou, Yaxin Liu, Yannie Liang, Anais White, Yunjie Li, Shreya Singh, Sanaz Bahargam, Mark Epstein, Sujoy Basu, Li Lao, Adnan Ozturel, Carl Crous, Alex Zhai, Han Lu, Zora Tung, Neeraj Gaur, Alanna Walton, Lucas Dixon, Ming Zhang, Amir Globerson, Grant Uy, Andrew Bolt, Olivia Wiles, Milad Nasr, Ilia Shumailov, Marco Selvi, Francesco Piccinno, Ricardo Aguilar, Sara McCarthy, Misha Khalman, Mrinal Shukla, Vlado Galic, John Carpenter, Kevin Villela, Haibin Zhang, Harry Richardson, James Martens, Matko Bosnjak, Shreyas Rammohan Belle, Jeff Seibert, Mahmoud Alnahlawi, Brian McWilliams, Sankalp Singh, Annie Louis, Wen Ding, Dan Popovici, Lenin Simicich, Laura Knight, Pulkit Mehta, Nishesh Gupta, Chongyang Shi, Saaber Fatehi, Jovana Mitrovic, Alex Grills, Joseph Pagadora, Tsendsuren Munkhdalai, Dessie Petrova, Danielle Eisenbud, Zhishuai Zhang, Damion Yates, Bhavishya Mittal, Nilesh Tripuraneni, Yannis Assael, Thomas Brovelli, Prateek Jain, Mihajlo Velimirovic, Canfer Akbulut, Jiaqi Mu, Wolfgang Macherey, Ravin Kumar, Jun Xu, Haroon Qureshi, Gheorghe Comanici, Jeremy Wiesner, Zhitao Gong, Anton Ruddock, Matthias Bauer, Nick Felt, Anirudh GP, Anurag Arnab, Dustin Zelle, Jonas Rothfuss, Bill Rosgen, Ashish Shenoy,

Bryan Seybold, Xinjian Li, Jayaram Mudigonda, Goker Erdogan, Jiawei Xia, Jiri Simsa, Andrea Michi, Yi Yao, Christopher Yew, Steven Kan, Isaac Caswell, Carey Radebaugh, Andre Elisseeff, Pedro Valenzuela, Kay McKinney, Kim Paterson, Albert Cui, Eri Latorre-Chimoto, Solomon Kim, William Zeng, Ken Durden, Priya Ponnapalli, Tiberiu Sosea, Christopher A. Choquette-Choo, James Manyika, Brona Robenek, Harsha Vashisht, Sebastien Pereira, Hoi Lam, Marko Velic, Denese Owusu-Afriyie, Katherine Lee, Tolga Bolukbasi, Alicia Parrish, Shawn Lu, Jane Park, Balaji Venkatraman, Alice Talbert, Lambert Rosique, Yuchung Cheng, Andrei Sozanschi, Adam Paszke, Praveen Kumar, Jessica Austin, Lu Li, Khalid Salama, Bartek Perz, Wooyeol Kim, Nandita Dukkipati, Anthony Baryshnikov, Christos Kaplanis, XiangHai Sheng, Yuri Chervonyi, Caglar Unlu, Diego de Las Casas, Harry Askham, Kathryn Tunyasuvunakool, Felix Gimeno, Siim Poder, Chester Kwak, Matt Miecnikowski, Vahab Mirrokni, Alek Dimitriev, Aaron Parisi, Dangyi Liu, Tomy Tsai, Toby Shevlane, Christina Kouridi, Drew Garmon, Adrian Goedeckemeyer, Adam R. Brown, Anitha Vijayakumar, Ali Elqursh, Sadegh Jazayeri, Jin Huang, Sara Mc Carthy, Jay Hoover, Lucy Kim, Sandeep Kumar, Wei Chen, Courtney Biles, Garrett Bingham, Evan Rosen, Lisa Wang, Qijun Tan, David Engel, Francesco Pongetti, Dario de Cesare, Dongseong Hwang, Lily Yu, Jennifer Pullman, Srini Narayanan, Kyle Levin, Siddharth Gopal, Megan Li, Asaf Aharoni, Trieu Trinh, Jessica Lo, Norman Casagrande, Roopali Vij, Loic Matthey, Bramandia Ramadhana, Austin Matthews, CJ Carey, Matthew Johnson, Kremena Goranova, Rohin Shah, Shereen Ashraf, Kingshuk Dasgupta, Rasmus Larsen, Yicheng Wang, Manish Reddy Vuyyuru, Chong Jiang, Joana Ijazi, Kazuki Osawa, Celine Smith, Ramya Sree Boppana, Taylan Bilal, Yuma Koizumi, Ying Xu, Yasemin Altun, Nir Shabat, Ben Bariach, Alex Korchemniy, Kiam Choo, Olaf Ronneberger, Chimezie Iwuanyanwu, Shubin Zhao, David Soergel, Cho-Jui Hsieh, Irene Cai, Shariq Iqbal, Martin Sundermeyer, Zhe Chen, Elie Bursztein, Chaitanya Malaviya, Fadi Biadsy, Prakash Shroff, Inderjit Dhillon, Tejasi Latkar, Chris Dyer, Hannah Forbes, Massimo Nicosia, Vitaly Nikolaev, Somer Greene, Marin Georgiev, Pidong Wang, Nina Martin, Hanie Sedghi, John Zhang, Praseem Banzal, Doug Fritz, Vikram Rao, Xuezhi Wang, Jiageng Zhang, Viorica Patraucean, Dayou Du, Igor Mordatch, Ivan Jurin, Lewis Liu, Ayush Dubey, Abhi Mohan, Janek Nowakowski, Vlad-Doru Ion, Nan Wei, Reiko Tojo, Maria Abi Raad, Drew A. Hudson, Vaishakh Keshava, Shubham Agrawal, Kevin Ramirez, Zhichun Wu, Hoang Nguyen, Ji Liu, Madhavi Sewak, Bryce Petrini, DongHyun Choi, Ivan Philips, Ziyue Wang, Ioana Bica, Ankush Garg, Jarek Wilkiewicz, Priyanka Agrawal, Xiaowei Li, Danhao Guo, Emily Xue, Naseer Shaik, Andrew Leach, Sadh MNM Khan, Julia Wiesinger, Sammy Jerome, Abhishek Chakladar, Alek Wenjiao Wang, Tina Ornduff, Folake Abu, Alireza Ghaffarkhah, Marcus Wainwright, Mario Cortes, Frederick Liu, Joshua Maynez, Andreas Terzis, Pouya Samangouei, Riham Mansour, Tomasz Kepa, François-Xavier Aubet, Anton Algymr, Dan Banica, Agoston Weisz, Andras Orban, Alexandre Senges, Ewa Andrejczuk, Mark Geller, Niccolo Dal Santo, Valentin Anklin, Majd Al Merey, Martin Baeuml, Trevor Strohman, Junwen Bai, Slav Petrov, Yonghui Wu, Demis Hassabis, Koray Kavukcuoglu, Jeff Dean, and Oriol Vinyals. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL https://arxiv.org/abs/2403.05530.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. Svd-llm: Truncation-aware singular value decomposition for large language model compression, 2024. URL https://arxiv.org/abs/2403.07378.

Yixuan Wang, Haoyu Qiao, Lujun Li, Qingfu Zhu, and Wanxiang Che. Commonkv: Compressing kv cache with cross-layer parameter sharing, 2025. URL https://arxiv.org/abs/2508.16134.

Haoyi Wu and Kewei Tu. Layer-condensed kv cache for efficient inference of large language models, 2024. URL https://arxiv.org/abs/2405.10637.

Huaijin Wu, Lianqiang Li, Hantao Huang, Tu Yi, Jihang Zhang, Minghui Yu, and Junchi Yan. HShare: Fast LLM decoding by hierarchical key-value sharing. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Tb5PY5vwp6.

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

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, Weijia Xu, Wenbiao Yin, Wenyuan Yu, Xiafei Qiu, Xingzhang Ren, Xinlong Yang, Yong Li, Zhiying Xu, and Zipeng Zhang. Qwen2.5-1m technical report, 2025a. URL https://arxiv.org/abs/2501.15383.

Haoqi Yang, Yao Yao, Zuchao Li, Baoyuan Qi, Guoming Liu, and Hai Zhao. Xquant: Achieving ultra-low bit kv cache quantization with cross-layer compression, 2025b. URL https://arxiv.org/abs/2510.11236.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering, 2024a. URL https://arxiv.org/abs/2405.15793.

Yifei Yang, Zouying Cao, Qiguang Chen, Libo Qin, Dongjie Yang, Hai Zhao, and Zhi Chen. Kvsharer: Efficient inference via layer-wise dissimilar kv cache sharing, 2024b. URL https://arxiv.org/abs/2410.18517.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. Turboquant: Online vector quantization with near-optimal distortion rate, 2025. URL https://arxiv.org/abs/2504.19874.

Jiamu Zhang, Liang Wu, Kelly Wan, Hanjie Chen, and Liangjie Hong. Spectra: Pushing the kv cache beyond the 2-bit cliff via spectral transform coding, 2026. URL https://arxiv.org/abs/2608.07915.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. $H_2o$: Heavy-hitter oracle for efficient generative inference of large language models, 2023. URL https://arxiv.org/abs/2306.14048.

Zhongzhu Zhou, Donglin Zhuang, Jisen Li, Ziyan Chen, Shuaiwen Leon Song, Ben Athiwaratkun, and Xiaoxia Wu. Oscar: Offline spectral covariance-aware rotation for 2-bit kv cache quantization, 2026. URL https://arxiv.org/abs/2605.17757.

Zayd Muhammad Kawakibi Zuhri, Muhammad Farid Adilazuarda, Ayu Purwarianti, and Alham Fikri Aji. Mlkv: Multi-layer key-value heads for memory efficient transformer decoding, 2024. URL https://arxiv.org/abs/2406.09297.

# A Related works

Work on the key-value cache divides by the object it acts on. One line keeps every entry and stores each number in fewer bits. A second line keeps full precision and stores fewer tokens. A third line changes what is stored at all and asks several layers to share one object. BellowsKV belongs to the third line, and this appendix places it against all three.

## A.1 Quantizing the cache

The most direct way to shrink a cache is to keep every entry and spend fewer bits on each number. What makes this hard is that the cache carries channel-aligned outliers, which are a few coordinates whose dynamic range forces a shared scale that crushes everything else. One group of methods protects those coordinates. KIVI (Liu et al., 2024b) quantizes keys along the channel axis and values along the token axis, so that the numbers sharing a scale are comparable to begin with, and it reaches two bits without any training. KVQuant (Hooper et al., 2024) calibrates a non-uniform grid per channel and keeps a sparse set of outliers in full precision, which is what allows it to report context lengths in the millions. OTT (Su et al., 2025a) tracks which tokens carry the outliers as decoding proceeds and stores those exactly, and it is the one baseline in our Protocol-A field that still works at two bits.

A second group removes the outliers instead of protecting them. QuaRot (Ashkboos et al., 2024) multiplies the cache by an orthogonal matrix before quantizing, so that every stored coordinate becomes a mixture of all the others and no single channel can dominate a scale. SpinQuant (Liu et al., 2025) learns that rotation rather than drawing it at random, and RotateKV (Su et al., 2025b) adapts it to the outlier structure of each head. PolarQuant (Han et al., 2025) changes the coordinate system instead of rotating inside it and codes each key by its direction and its magnitude, while TurboQuant (Zandieh et al., 2025) replaces scalar rounding with online vector quantization at a near-optimal distortion rate. The most recent work treats the whole problem as transform coding and fits the transform to the model's own activation statistics offline, which is the shared idea behind OSCAR (Zhou et al., 2026), OScaR (Su et al., 2026) and SPECTRA (Zhang et al., 2026), and Staniszewski & Łańcucki (2026) state the coding view explicitly. Every method in this paragraph acts inside a single layer, and every one of them applies its transform again at each read. The latent BellowsKV stores is already a dense mixture of the channels of several layers, so it needs no rotation before storage, and Section 4.3 tests that claim against each of these methods whose released code we could run.

## A.2 Evicting tokens

The second line keeps the stored numbers exact and drops tokens instead. H2O (Zhang et al., 2023) observes that a small set of heavy hitters receives most of the attention mass over a long prefix and keeps those tokens. SnapKV (Li et al., 2024) scores each cached token by the attention that the most recent queries place on it and then pools the profile over a short window, so that a token kept for its own score also keeps its neighbours. StreamingLLM (Xiao et al., 2024) shows that the first few positions act as attention sinks and must never be dropped, which is why nearly every later method, including ours, exempts a sink region and a recent window. Eviction is the one axis whose damage cannot be undone at read time, because a token that has been dropped is gone for every later query.

All of these methods decide layer by layer, since each layer owns its cache and is free to rank tokens as it likes. A smaller body of work asks whether that freedom is worth its cost. Ma et al. (2025) report that neighbouring layers attend to much the same tokens and reuse one layer's decision in the next, and HShare (Wu et al., 2025) organises the same redundancy hierarchically across layers, heads and queries in order to speed up decoding. Our eviction operator takes this reading to its conclusion. A block of layers holds one stored object, so a token admits exactly one decision, and Section 4.3 prices that single decision against the strongest per-layer control we could build at a matched byte budget.

### A.3 SHARING THE CACHE ACROSS LAYERS

The third line changes the stored object itself, and it appears first in architectures that are trained to share. Cross-layer attention (Brandon et al., 2024) lets a group of layers read one cache, YOCO (Sun et al., 2024) caches once for an entire decoder, MLKV (Zuhri et al., 2024) shares key and value heads across layers, and LCKV (Wu & Tu, 2024) condenses the caches of many layers into one. Together they establish that a block of layers can live with a single stored object, but they reach it by retraining the attention stack on billions of tokens, which is not available to a checkpoint that is already deployed. The same holds along the head axis, where GQA (Ainslie et al., 2023) and multi-head latent attention (DeepSeek-AI et al., 2024) are architectural decisions taken before training, although recent work converts a trained model to latent attention after the fact (Meng et al., 2025; Ji et al., 2025).

Training-free cross-layer compression is the direction this paper works in, and it is recent. xKV (Chang et al., 2026) factorizes the caches of a window of layers into one shared low-rank basis that it recomputes for every sequence at prefill. CommonKV (Wang et al., 2025) derives a shared projection from the grouped layers' own weight matrices and then merges the token slots whose latents agree across the group. MiniCache (Liu et al., 2024a) merges adjacent layers by interpolating per-token unit vectors while each layer keeps its own norm, KVSharer (Yang et al., 2024b) searches for the layer pairs whose caches are most dissimilar and lets the later layer reuse the earlier one, and XQuant (Yang et al., 2025b) reuses quantization codes across layers at very low bit-widths. The single-layer limit of the same construction is a per-layer low-rank cache, which Palu (Chang et al., 2024) and Eigen Attention (Saxena et al., 2024) build on the activation-aware whitening of SVD-LLM (Wang et al., 2024), and Section 3.2 shows that our objective reduces to exactly that construction once the cross-layer blocks of the second-moment matrix are set to zero. A recent position paper argues that quantization dominates rank reduction at any matched budget (Salfati, 2026). Our measurements agree with it in the sense that matters here, because the axis that pays is the number of layers one stored object serves and not the rank at which that object is stored.

## B EXTENDED METHOD

Main Section 3 states the method; this appendix carries the proof, the gauge analysis, the full whisper and decode detail, the accounting, and the negative results.

### B.1 PROPOSITION 1: OPTIMALITY OF THE CALIBRATED CONSTRUCTION

The objective of Eq. (2) is a reduced-rank regression (Izenman, 1975) with the error measured on the keys rather than on the weights.

> **Proposition 1.** *Let* $\mathbf{G} = \mathbb{E}[\mathbf{H}^\top \mathbf{H}] \succ 0$ *(enforced by a small ridge* $\mathbf{G} \leftarrow \mathbf{G} + \varepsilon \bar{g} \mathbf{I}$*, as in Wang et al., 2024), let* $\mathbf{M} := \mathbf{G}^{1/2}\mathbf{W}$ *have thin singular value decomposition* $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$*, and let* $R \le \operatorname{rank}(\mathbf{M})$*. The minimizers of Eq. (2) are exactly* $\mathbf{W}_\downarrow^\star = \mathbf{G}^{-1/2}\mathbf{U}_{:R}\mathbf{T}$ *and* $\mathbf{W}_\uparrow^\star = \mathbf{T}^{-1}\mathbf{\Sigma}_{:R}\mathbf{V}_{:R}^\top$ *for any invertible* $\mathbf{T} \in \mathbb{R}^{R\times R}$*, with minimum value* $\sum_{i>R}\sigma_i^2$*.*

**Proof.** (1) For any $\mathbf{X}$, $\mathbb{E}\|\mathbf{H}\mathbf{X}\|_F^2 = \operatorname{tr}(\mathbf{X}^\top\mathbf{G}\mathbf{X}) = \|\mathbf{G}^{1/2}\mathbf{X}\|_F^2$; with $\mathbf{X} = \mathbf{W} - \mathbf{W}_\downarrow\mathbf{W}_\uparrow$ the objective becomes the deterministic $\|\mathbf{G}^{1/2}(\mathbf{W} - \mathbf{W}_\downarrow\mathbf{W}_\uparrow)\|_F^2$. (2) $\mathbf{G}^{1/2}$ is invertible; substituting $\mathbf{A} := \mathbf{G}^{1/2}\mathbf{W}_\downarrow$ gives $J = \|\mathbf{M} - \mathbf{A}\mathbf{W}_\uparrow\|_F^2$. (3) $\mathbf{A}\mathbf{W}_\uparrow$ ranges over exactly the matrices of rank at most $R$, so the problem is unconstrained best rank-$R$ approximation of $\mathbf{M}$; by the Eckart–Young–Mirsky theorem (Eckart & Young, 1936; Mirsky, 1960) the minimizer is $\mathbf{U}_{:R}\mathbf{\Sigma}_{:R}\mathbf{V}_{:R}^\top$ with residual $\sum_{i>R}\sigma_i^2$. (4) Any rank-$R$ factorization of that minimizer is unique up to an invertible right action $\mathbf{T}$ on $\mathbf{A}$; un-whitening gives the stated pair. ■

Setting the cross-layer blocks of $\mathbf{G}$ to zero decouples the problem into $n$ per-layer regressions, the per-layer channel-axis construction of Zhang et al. (2026), which rests on the activation-aware whitening of Wang et al. (2024). The primitive is therefore not ours; the off-diagonal blocks $\mathbf{G}_{ij} = \mathbb{E}[\mathbf{h}_{\ell_i}\mathbf{h}_{\ell_j}^\top]$ are the added, load-bearing ingredient, and a per-layer method structurally cannot

use them. Values use an independent pair $(\mathbf{W}_{\downarrow}^{V}, \mathbf{W}_{\uparrow}^{V})$ obtained identically; biases are added after reconstruction.

### B.2 The gauge, and why the shipped split is balanced

$\mathbf{T}$ is invisible at full precision, since every choice spans the same subspace at the same minimum, and matters only once the latent is quantized, because it reshapes the latent's coordinate statistics. Two named choices. *Whiten-flat* ($\mathbf{T} = \mathbf{I}$): all singular-value energy sits in the fixed up-projection and the cached latent is exactly isotropic, $\mathbb{E}[\mathbf{c}\mathbf{c}^{\top}] = \mathbf{U}_{:R}^{\top}\mathbf{G}^{-1/2}\mathbf{G}\mathbf{G}^{-1/2}\mathbf{U}_{:R} = \mathbf{I}_R$. *Balanced* ($\mathbf{T} = \mathbf{\Sigma}_{:R}^{1/2}$; shipped, and also the split of Wang et al., 2025): $\mathbb{E}[\mathbf{c}\mathbf{c}^{\top}] = \mathbf{\Sigma}_{:R}$, an energy-ordered latent.

The energy-ordered latent looks like what uniform quantization fears, coordinate variances spanning orders of magnitude, but a shared scale is defeated by *within-group* heterogeneity, not by a smooth global slope. The quantizer shares one scale per 128 adjacent coordinates; in the energy-ordered latent the spectrum decays smoothly, so each group is internally homogeneous and the per-group min–max scales absorb the ordering across groups: ordering plus grouping acts as a free piecewise variance-adaptive quantizer. Measured at the shipped operating point the balanced split is ahead of whiten-flat by a small margin (about 1.7 retention points on average), and we keep whiten-flat as an ablation arm. Two honesty notes. The margin is small, and BellowsKV's low-bit robustness traces to the fixed activation-weighted basis, not to $\mathbf{T}$. And the isotropy identity stays an analysis device: equal-variance coordinates would matter under single-scale quantization, which BellowsKV does not use.

### B.3 Operating points and accounting

Uniform $b$-bit quantization is applied per token, per 128-channel group, asymmetric min–max (Liu et al., 2024b), with a $0.95$ clip on each group's range at the high tier only (it hurts below, Appendix B.9). The rank stays full ($R{=}p$) at both tiers; a remainder block shorter than $n$ is kept, never left uncompressed.

| tier | block | bits | clip | per-token-per-block cost (bits) |
|---|---|---|---|---|
| low ($\sim$4×) | $n{=}2$ | INT8 | 1.0 | $(8R + \frac{R}{4} + 512) + (8R + \frac{R}{4})$ |
| high ($\sim$16×) | $n{=}4$ | INT4 | 0.95 | $(4R + \frac{R}{4} + 512) + (4R + \frac{R}{4})$ |
| top ($\sim$18×) | $n{=}4$ | K3V2 | 0.95 | $(3R + \frac{R}{4} + 512) + (2R + \frac{R}{4})$ |

The two brackets are the key and value sides; $R/4$ is each side's fp16 group scale-and-zero, and $512$ is the key side's whisper coefficients ($r{=}32$ at fp16, Appendix B.6). Every reported ratio counts all of it plus the fp16 window of Appendix B.5. Measured at 32k context: $3.64\times$ and $13.84\times$ on Llama and Mistral, $3.53\times$ and $13.13\times$ on Qwen (KV width 512); the high tier at INT3 measures $17.65\times$ ($16.52\times$ on Qwen) and at K3V2, $(3R + \frac{R}{4} + 512) + (2R + \frac{R}{4})$, $20.47\times$ ($18.96\times$ on Qwen). Under Protocol A's shorter mixed-length inputs the window's fixed cost weighs more and the measured ratios are correspondingly lower; the Protocol-A tables state their own values. Dropping the whisper and window terms recovers the bare formula $2(bR + R/4)$.

### B.4 The eviction verdict

BellowsKV-Evict is the block-level form of attention-mass scoring (Zhang et al., 2023; Li et al., 2024); every setting below is fixed once and used unchanged on all three models.

**Scoring.** After prefill, the queries of the last $64$ prompt positions attend to every cached position, and each cached token receives the attention mass these queries place on it, summed over the block's $n$ layers and all their heads. The scores are computed on the keys the block will actually read at decode, reconstructed from the quantized latent through the read-heads and whisper-corrected, not on the exact prefill keys, so the verdict sees exactly what decode will see. The summed profile is then max-pooled over a width-13 window along the sequence, so that a token kept for its own score also keeps its neighbours (Li et al., 2024).

**Selection.** The first $4$ positions and the last $64$ prompt positions are exempt and always kept (Appendix B.5). Of the remaining positions, the fraction $q$ with the highest pooled scores survives; the rest are dropped once for the whole block, a single gather applied to the latent and, in the composed operator, to its group scales and whisper coefficients alike. Because the block holds one slot per token, the gather is issued once per block rather than once per layer, so the verdict costs $1/n$ of its per-layer counterpart and is consistent across the block by construction.

**Accounting.** The keep ratio $q$ is applied to the compressed cache, so the stored size per block is $q$ times the per-token cost of Appendix B.3 plus the exempt window. BellowsE alone prunes the $n{=}4$ fp16 latent; BellowsQ×BellowsE prunes the INT4 tier. At the paired points of Table C.4 BellowsE stores 1–5% more than the per-layer control at the same nominal budget, which is the direction that disfavours it.

**The per-layer control.** The control runs the identical scoring and pooling, but each layer scores on its own exact fp16 cache and keeps its own top fraction, so its verdicts differ layer to layer and its gather is issued $n$ times. The naive stack of Section 4.3 (xKV + INT4 + SnapKV) needs a verdict pooled per xKV group for the same reason BellowsE does: per-layer verdicts cannot share one stored basis.

### B.5 The recency window

The first $4$ positions and the last $64$ prompt positions keep their original per-layer fp16 tensors, and decode-time tokens are appended per-layer as usual, so the recent region is exact at all times: the standard residual-window practice (Liu et al., 2024b; Xiao et al., 2024) applied at the block level. Cost: $68$ prompt positions, $0.2\%$ of a 32k context (charged in Appendix B.3). What it buys concentrates where the codec is weakest: recency-dominated tasks such as code completion and long-form generation recover 5–11 raw points at the high tier, while retrieval-style tasks are unchanged.

### B.6 The whisper in full

**Construction.** At prefill the true keys are still in hand, so the residual $\mathbf{E} = \mathbf{K} - \hat{\mathbf{K}}$ (pre-rotation, bias-free) is free to read. A randomized range finder (Halko et al., 2010) sketches its top $r{=}32$ right-singular directions: draw a Gaussian $\mathbf{\Omega} \in \mathbb{R}^{np\times(r+8)}$, form $\mathbf{E\Omega}$, orthonormalize, project, and take the top-$r$ right singular vectors $\mathbf{V}_r$; three matrix products and one tiny SVD, milliseconds per block. Stored: $\mathbf{V}_r$ once per sequence ($np \times 32$ at fp16, about $0.7\%$ of the cache at 32k) and 32 fp16 coefficients per token (the $512$ bits of Appendix B.3). Every later key read adds back $\mathbf{C}_t\mathbf{V}_r$, one rank-32 product. Measured, $\|\mathbf{E}\|$ is roughly $10\%$ of $\|\mathbf{K}\|$ and the sketch captures about three quarters of it.

**What it buys.** On averages the whisper is nearly free-standing: removing it costs about $0.1$ mean retention points at the high tier ($97.9$ against $98.0$, Llama, Protocol B, Table 20), and at the fp16 sharing-only points the paired no-whisper arms sit $0.05$–$0.4$ points lower. Its value is in the tail: the largest per-task swing is $4$ points (2WikiMQA, $101.5$ with the whisper against $97.5$ without). The whisper is a worst-case stabilizer, not an average lifter, and we claim no more for it.

**Keys only, and why.** A key error passes through the softmax and perturbs *which* positions are attended, a brittle failure worth correcting everywhere. A value error enters the convex combination $\sum_t a_t \mathbf{v}_t$ and averages out over a long prefill. Patching values helps exactly when the prefill is long (the +V-whisper row of Table 20: $98.7$ against $98.0$ at 32k) and flips sign on short-prefill, generation-heavy inputs, where the rank-32 structure perturbs the few vectors that dominate the average. One rule is shipped uniformly: whisper the keys, leave the values to the attention average.

### B.7 Decode paths

**Mode A (shipped): values absorbed, keys reconstructed.** Values carry no positional rotation, so the read-head commutes with the attention-weighted sum,

$$\sum\nolimits_t a_t \left(\mathbf{W}_{\uparrow}^{V,(i)\top} \mathbf{c}_t^V\right) = \mathbf{W}_{\uparrow}^{V,(i)\top} \Big(\sum\nolimits_t a_t\, \mathbf{c}_t^V\Big), \tag{4}$$

the weight-absorption identity of multi-head latent attention (DeepSeek-AI et al., 2024): every layer attends directly in the $R$-dimensional latent, $\mathbf{W}_{\uparrow}^{V,(i)}$ folds into the output projection, and the value

path is exact up to the codec itself. Keys cannot use the identity because the positional rotation sits between the latent and the score: each key is reconstructed through its read-head, whisper-corrected, rotated per head, and attended, a rank-$R$ plus rank-32 product per layer that hides under the memory-bound cache read.

Table 6: **BellowsKV decodes at parity, and the same relation prices Palu's speedup within 6%.** Decode speed and capacity (H100 SXM, HuggingFace Transformers, 32k context, no custom kernel): the reconstruct-at-read relation of Section 4.4 on both designs it covers, and what the freed bytes buy under two engines.

| Design | Read path | Relation predicts | Measured / reported |
|---|---|---|---|
| | *decode speed* | | |
| BellowsKV | $R{=}d_{kv}$, fp16 | 1.00× | 0.995× (32.5 tok/s) |
| Palu (rank-cutting) | $R{=}d_{kv}/2$, fp16 | 2.00× | 1.89× (reported) |
| | *capacity from the freed bytes* | | |
| HuggingFace Transformers | max batch | – | 2× |
| vLLM paged allocator | KV pool | – | 14.76× (6.02M tokens) |

**Mode B (optional restructuring, not a speedup).** An MLA-style decoupled-rotation split of the key path (DeepSeek-AI et al., 2024; Meng et al., 2025; Ji et al., 2025) can move the score into the latent space. The relation of Section 4.4 shows why we do not ship it: at $R = d_{kv}$ consumed in fp16 both of its factors are one, so absorption removes arithmetic but not bytes, and measured decode is $0.995\times$ dense either way. Only a smaller $R$ or a sub-fp16 consumption format changes the bound, and those are exactly the moves the accuracy story forbids or defers.

### B.8 Costs

Calibration is one-shot: one forward pass over 32 windows of 2048 tokens (half WikiText-2, half code) accumulates the second-moment blocks, then one singular value decomposition per block; about twenty minutes on one GPU per model, no gradients, and the factors fold into the weights. We fix this one corpus recipe globally rather than tuning the mix per model or task. Per sequence, the only added computation is the whisper sketch (milliseconds per block, roughly two orders of magnitude below xKV's per-sequence full SVD; a fully per-sequence variant of our own construction costs $2.2\times$ the prefill and scores worse, Appendix B.9). Decode adds one dequantization per block and the products of Appendix B.7.

### B.9 Negative results

Each of these was implemented, run, and rejected; they bound what the ingredients do and do not buy.

1. **Sharing reconstructed keys across layers.** Token-wise cosine between adjacent-layer keys is near zero ($-0.002$ on Qwen, $+0.0016$ on Llama), and substituting one shared key across even two layers inflates perplexity by three orders of magnitude under an oracle choice of the shared key (diagnostic probe). Sharing must happen one step earlier, in the latent; this is the design fact behind Section 2.1.
2. **Quantization-aware read-head refit** (ridge refit of $\mathbf{W}_{\uparrow}$ against the quantized latent): flat to negative on probe cells. Proposition 1's solution is already near-optimal, and INT4 noise on the latent is effectively unbiased for the objective.
3. **Per-sequence gauge** (whitening the latent per sequence around the quantizer): 96.7 against the shipped 98.0 mean retention. The per-group scales already deliver the adaptation, and the sequence-level estimate adds variance.
4. **Fully per-sequence basis** (this sequence's optimal rank-$R$ SVD in place of the calibrated consensus): 97.3 at $2.2\times$ prefill cost, below the frozen consensus. Test-time subspace

adaptation is not the active ingredient; a stable average basis plus a rank-32 correction beats a fully sequence-specific basis.

5. **V-whisper as a default**: the sign flips with prefill length (Appendix B.6); an ablation row, not the method.
6. **Whiten-flat split as the shipped gauge**: behind balanced by about 1.7 points at the operating point (Appendix B.2); kept as the ablation arm.
7. **Clip below the high tier**: the 0.95 range clip helps only at $n$=4×INT4; at INT8 it hurts and is disabled.
8. **Depth-adaptive rank allocation and non-adjacent groupings**: both flat to negative in probes; consecutive blocks at uniform full rank remain the default.

# C Extended experiments

## C.1 Setup, accounting, and coverage

**The two protocols.** Every experiment runs under one of two protocols, one per baseline family, and no number crosses between them.

| | **Protocol A** | **Protocol B** |
|---|---|---|
| question | absolute accuracy under compression | fraction of dense accuracy retained |
| harness | official LongBench pipeline | the harness released by the xKV authors |
| inputs | 8 English tasks, mixed lengths | 11 tasks, only inputs longer than 4k tokens (up to the 32k window) |
| unit | score 0–100 | retention % = score ÷ dense score |
| compared against | the quantization family | the cross-layer and eviction families |
| sample | full test sets (200/500 per task) | all qualifying samples (Table 7) |

Retention is computed per task against the dense run of the same harness and campaign; values above 100% occur and are expected. Task codes: Qsp (Qasper), MFQ (MultiFieldQA-en), HQA (HotpotQA), MSQ (MuSiQue), NQA (NarrativeQA), Gov (GovReport), QMS (QMSum), PsR (Passage Retrieval-en), LCC, RBP (RepoBench-P), 2Wk (2WikiMultihopQA).

Table 7: **Protocol B qualifying sample per task** (inputs longer than 4k tokens and within the 32k window; the count differs per model because tokenizers differ). Every Protocol-B number in the paper uses all qualifying samples.

| Model | Qsp | MFQ | HQA | MSQ | NQA | Gov | QMS | PsR | LCC | RBP | 2Wk |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Llama-3.1-8B | 122 | 110 | 195 | 200 | 115 | 177 | 199 | 200 | 93 | 442 | 175 |
| Mistral-7B | 143 | 117 | 195 | 200 | 85 | 182 | 190 | 200 | 150 | 470 | 183 |
| Qwen2.5-7B | 125 | 111 | 195 | 200 | 115 | 180 | 199 | 200 | 96 | 444 | 178 |

**BellowsKV configuration.** One codec serves every table: balanced split, 32 calibration windows, full rank, sink 4 + recent 64, key whisper $r$=32. BellowsQ ships two tiers, $n$=2 INT8 (no clip) and $n$=4 INT4 (clip 0.95); the high tier is also stored at INT3 and at K3V2 (key latent 3 bits, value latent 2 bits) with the same codec and clip, and the ∼8× and ∼12× appendix tiers use $n$=2 and $n$=3 at INT4. BellowsE is the $n$=4 fp16 latent pruned by the verdict of Section 3.3; BellowsS is that latent with neither operator. The accounting formula and the measured per-model ratios are in Appendix B.3; every reported ratio counts all metadata.

**Compression tiers.** Every comparison table groups its rows into tiers of *measured* compression ratio (Table 8), sorts a tier by ratio, and marks its best and second average. A row belongs to the tier its own measured ratio falls in, never to a nominal one, so two rows between the same pair of rules are the rows to compare, and the Comp. column shows how close their bytes are. The bins are fixed once for each axis and used unchanged in the main text and the appendix.

Table 8: **Compression tiers.** Bins on the measured ratio (all metadata counted) that every comparison table uses; a row belongs to the tier its own measured ratio falls in, never to a nominal one. The edges are chosen so that the highest ratio inside a tier is always a BellowsKV point, which puts every within-tier comparison on the baselines' side of the byte budget, and so that no tier holds our rows alone. Protocol A's top tier is the one exception: no per-layer quantizer exists past $8\times$.

| Axis | Tier | Measured ratio |
|---|---|---|
| Protocol A (quantizers) | $<4\times$ | $<4\times$ |
| | $\sim5\times$ | 4–6.6$\times$ |
| | $\sim8\times$ | 6.6–9.5$\times$ |
| | $\geq12\times$ | $\geq9.5\times$ (no per-layer point exists) |
| Protocol B (sharing) | $\leq4\times$ | 1–3.9$\times$ (as-shipped fp16 baselines) |
| | 4–11$\times$ | 3.9–10.9$\times$ |
| | $\sim12\times$ | 10.9–14$\times$ |
| | $\geq14\times$ | 14–24$\times$ |
| Eviction (budget) | $\sim2\times$ | 1.5–3$\times$ |
| | $\sim4\times$ | 3–6$\times$ |
| | $\sim8\times$ | 6–10$\times$ |
| | $\sim16\times$ | 13–20$\times$ |
| | $\sim30\times$ | 25–35$\times$ |
| | $\sim57\times$ | 50–60$\times$ |

**Baseline configurations.** Tables 9 and 10 give every baseline configuration behind the numbers in this paper: the Protocol-A quantizers at their published settings, and the Protocol-B cross-layer methods at the settings that produce each baseline row of Table 1. Protocol-A baselines run each method's own official code, with group size 128 and each method's own residual-window convention. Protocol-B baselines all run inside the xKV harness on the same qualifying samples against the same dense run.

Table 9: **Every Protocol-A baseline configuration, and the models it exists for.** Each method runs its own official code at its published settings, with group size 128 and the method's own residual-window convention. ✓ marks a model the method ships modelling for; coverage on Qwen is thinner than on the other two, which is stated wherever it matters. †RotateKV's 4-bit configuration exists only in the Mistral config set. ‡Neither PolarQuant nor TurboQuant ships a Qwen path, so the Qwen adapters are ours and quantize bias-free keys (see *Compatibility fixes*).

| Method | Configurations run | Llama | Mistral | Qwen |
|---|---|---|---|---|
| KIVI | K4V4, K2V2 | ✓ | ✓ | – |
| OTT | K4V4, K2V2 | ✓ | ✓ | – |
| RotateKV | 4-bit†, 3-bit, 2-bit | ✓ | ✓ | – |
| TurboQuant | K4V4, K4V2, K2V2 | ✓ | ✓ | ✓‡ |
| PolarQuant | r4t4v4, r3t3v4, r2t2v4, r2t2v2 | ✓ | – | ✓‡ |

Table 10: **The setting behind every Protocol-B baseline row of Table 1, with the ratio it measures.** All of them run inside the xKV harness on the same qualifying samples against the same dense run; the four training-free methods other than xKV were ported into it for this paper and the harness patches are released with the code. Where a setting differs on Qwen, whose KV width is half and which has 28 layers, the Qwen value is in parentheses; where all three differ the values are listed Llama, Mistral, Qwen. $p$ is xKV's full per-sequence rank. Storage bits are the method's own composition, not ours.

| | | measured compression | | |
|---|---|---|---|---|
| Method | Setting | Llama | Mistral | Qwen |
| xKV | rank $p$, fp16 | 4.00× | 4.00× | 4.00× |
| xKV | rank $p/2$, fp16 | – | – | 8.00× |
| xKV | rank $p/4$, fp16 | 16.00× | 16.00× | 16.00× |
| CommonKV | $r$=3328 ($r$=1664), INT4 | 3.95× | 3.95× | 3.95× |
| CommonKV | $r$=1664 ($r$=832), INT4 | 7.89× | 7.90× | 7.89× |
| CommonKV | $r$=832 ($r$=416), INT4 | 15.78× | 15.80× | 15.78× |
| MiniCache fp16 | deeper half, fp16 | 1.31× | 1.31× | 1.31× |
| MiniCache+INT4 | deeper half, INT4 | 4.78× | 4.78× | 4.78× |
| MiniCache all+INT4 | every pair, INT4 | 6.72× | 6.72× | 6.72× |
| MiniCache+INT2 | every pair, INT2 | 11.96× | 11.99× | 11.97× |
| KVSharer fp16 | 8/32 shared (7/28), fp16 | 1.33× | 1.33× | 1.33× |
| KVSharer+INT4 | 8/32 shared (7/28), INT4 | 4.99× | 4.99× | 4.99× |
| KVSharer 37.5%+INT2 | 12/32 shared, INT2 | – | 11.24× | – |
| KVSharer+INT2 | 12/32 / 16/32 / 10/28 shared, INT2 | 11.24× | 14.05× | 10.93× |
| XQuant published | classical schedule | 6.59× | 6.59× | 6.59× |
| XQuant 1-bit | 1-bit everywhere, groups 64 | 15.11× | 15.11× | 15.11× |

**Compatibility fixes.** Three baselines needed a change before they would run at all, and in each case the change removes a handicap rather than adding one. PolarQuant's released code ships only its key-side path, which leaves values in fp16 at about 1.6×, with the Mixed K+V path present but commented out; we enable it, because the key-only path cannot reach the operating points its own paper reports. On Qwen, PolarQuant and TurboQuant quantize bias-free keys, that is $k$ minus the rotated $k$-projection bias, added back at read time; the bias is a weight, so the ratio is unchanged. Qwen2.5's key bias dominates the key norm, and both methods code direction and magnitude, so on raw keys they collapse and score higher at fewer bits; those raw-key runs are kept invalidated and are not reported. Neither method ships a Qwen path, so the Qwen adapters are ours, as are CommonKV's Qwen merge weights, which we recomputed with the authors' own sensitivity procedure, the perplexity increase when one layer's keys are randomized, normalized within each group; it reproduces their shipped Llama list to a correlation of 0.99. MiniCache released no code, and its printed inequality would retain the least divergent token rows rather than the most divergent ones, so we follow its prose and its 5% budget.

**Accounting.** Two baselines are charged differently from their papers, in both cases because the published figure omits bytes the method actually stores. CommonKV's official code installs the same up-projection for keys and values, so a layer's key latent and value latent are one $r$-dimensional vector: we store and charge it once, and we credit its cross-layer token merge, under which the 40% of token slots whose first- and last-layer latents are least similar stay per-layer, together with the first 4 and last 128 positions, while every other slot holds one weighted average per group. Its rank is then set so that this honest ratio lands on the tier. XQuant's published "1.375-bit" figure excludes scales and zeros, which at group 32 add a full bit per value; our ratio counts them and the 128-token residual.

**Coverage.** Protocol-A coverage is uneven because KIVI, OTT and RotateKV ship no Qwen modelling and PolarQuant ships no Mistral modelling; Table 9 states which rows exist, and Qwen's Protocol-A field is thinner than the other two models' wherever it matters. On the Protocol-B side,

xKV's Qwen runs at rank $p$ and $p/2$ lack the MuSiQue cell, so their averages are over ten tasks and carry a † wherever they appear. KVSharer's Qwen rows are low at every setting because its own strategy search makes layer 0 the donor for layers 1–3, its most dissimilar pairs, and the model does not tolerate that at long inputs. For every Protocol-B row, sharing is applied to the cache read at decode and prefill attention is exact.

Table 1 in the main text is the complete Protocol-B picture: every operating point of every method on every model, tier by tier (bins in Table 8), including the baselines' as-published fp16 settings and every low-bit setting they compose with. KVSharer's largest shared set with INT2 lands at 14.1× on Mistral and near 11× on Llama and Qwen, so the same setting sits in different tiers on different models; the table follows the measured ratio. Figures 3–5 draw the same rows as a frontier, one figure per model: retention against measured compression, one line per method.

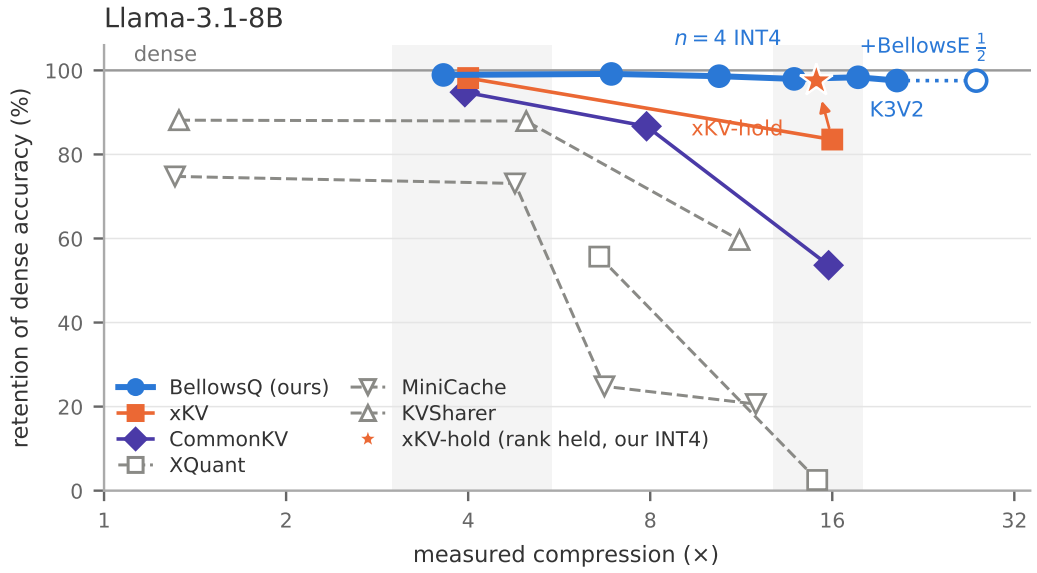



Figure 3: **On Llama the cross-layer axis stalls at 16× because rank is cut, and holding the rank recovers it.** Protocol B retention of dense accuracy against measured compression, one line per training-free cross-layer method over all of its registered operating points, one harness, full sample. Filled markers are the latent methods and hollow grey markers the raw-cache methods; the shaded bands are the ∼4× and ∼16× tiers of Table 1. The star is xKV-hold, xKV's own basis kept at near-full rank and stored with our INT4 leg instead of truncated (Section 4.2); the hollow blue circle is BellowsQ composed with a keep-1/2 verdict (Section 4.3). †10 tasks.

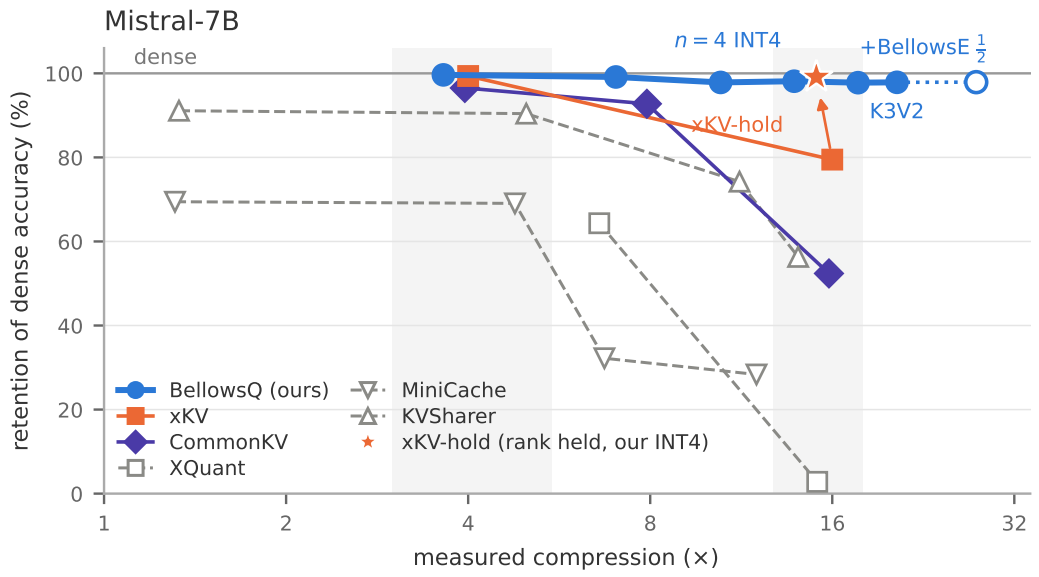



Figure 4: **Mistral repeats the pattern: every latent method works at ∼4×, only the full-rank latent survives past 12×, and the transplant recovers xKV to within a point of the consensus.** Axes, series and markers as in Figure 3.

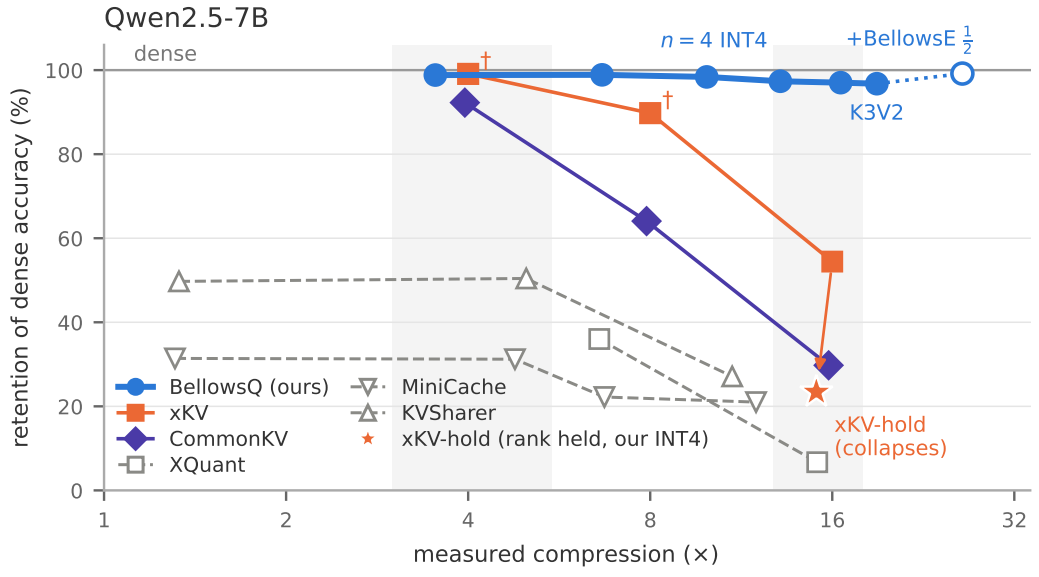



Figure 5: **Qwen is where the rivals fall furthest and where the transplant collapses.** xKV loses more than forty points between 4× and 16×, and xKV-hold, which recovers on the other two models, drops to 23.4% here, for a cause we have not isolated (Section 4.2). Axes, series and markers as in Figure 3.

**The CommonKV numbers.** Under the accounting above CommonKV is a working 4× method. An earlier version of this comparison, taken from its own harness, charged two latents per layer and gave the merge no credit, which placed its 4× point at a rank where the method has already collapsed. What the corrected rows then test is whether the axis extends, which is this paper's

question, and the mechanism at $16\times$ is the one Section C.6 prices: the per-layer latent must fall to a rank below the viability floor before the bytes are saved.

### C.2 Protocol A: full leaderboards

Tables 11–13 give the complete Protocol-A picture per model, tier by tier: every BellowsKV configuration from $1.92\times$ to $18.6\times$ (BellowsS rows are the sharing-only fp16 latent; BellowsQ rows add uniform bits, and the K3V2 row keeps the key latent at 3 bits while storing the value latent at 2) against every official baseline configuration. The $\geq 12\times$ tier holds our rows alone, the one tier in the paper that does: a per-layer quantizer at full rank bottoms out at 2 bits, which is $8\times$, and going further forces rank truncation, the regime Section C.6 shows collapsing, so no per-layer point exists there to compare with. Every other tier, here and in Table 1, holds at least one baseline and has a BellowsKV point as its highest ratio (Table 8).

Table 11: **Protocol A, Llama-3.1-8B.** Absolute LongBench scores, full test sets, every BellowsKV point and every official baseline configuration, grouped by measured compression tier (rules separate tiers; bins in Table 8); within a tier rows are sorted by ratio and the **best** and second Avg are marked. Baselines run each method's official code; PolarQuant runs its Mixed K+V path, and on Qwen it and TurboQuant quantize bias-free keys (Appendix C.1).

| Method | Comp. | Qasper | QMSum | MNews | TREC | TQA | SAMSum | LCC | RB-P | Avg |
|---|---|---|---|---|---|---|---|---|---|---|
| Dense fp16 | 1.00× | 46.57 | 25.19 | 27.03 | 72.50 | 91.65 | 43.47 | 63.38 | 56.55 | **53.29** |
| BellowsS $n$=2 | 1.95× | 46.76 | 25.32 | 26.08 | 72.50 | 91.28 | 43.90 | 62.38 | 56.32 | 53.07 |
| BellowsS $n$=3 | 2.90× | 45.25 | 25.13 | 25.90 | 72.50 | 91.44 | 43.69 | 62.28 | 55.15 | 52.67 |
| PolarQuant r4t4v4 | 3.66× | 46.01 | 25.56 | 26.69 | 72.50 | 91.86 | 42.88 | 65.38 | 58.44 | 53.66 |
| BellowsQ $n$=2 INT8 | 3.67× | 46.47 | 25.39 | 26.19 | 72.50 | 91.28 | 43.77 | 62.52 | 56.28 | 53.05 |
| **KIVI K4V4** | 3.76× | 46.11 | 25.14 | 27.11 | 72.50 | 92.42 | 44.10 | 65.05 | 58.91 | **53.92** |
| OTT K4V4 | 3.76× | 46.47 | 25.30 | 26.66 | 72.50 | 91.73 | 43.73 | 64.96 | 59.28 | 53.83 |
| BellowsS $n$=4 | 3.84× | 46.30 | 24.97 | 25.26 | 72.50 | 91.49 | 44.70 | 61.23 | 55.80 | 52.78 |
| TurboQuant K4V4 | 4.00× | 45.87 | 25.21 | 26.79 | 73.00 | 91.06 | 43.68 | 62.44 | 54.65 | 52.84 |
| **PolarQuant r3t3v4** | 4.13× | 45.60 | 25.38 | 26.60 | 72.50 | 92.16 | 42.44 | 65.36 | 57.97 | **53.50** |
| PolarQuant r2t2v4 | 4.74× | 42.52 | 24.37 | 25.46 | 66.00 | 91.50 | 40.91 | 61.13 | 53.18 | 50.63 |
| RotateKV 3-bit | 5.33× | 45.67 | 24.69 | 26.85 | 73.50 | 90.34 | 42.86 | 61.42 | 53.71 | 52.38 |
| TurboQuant K4V2 | 5.33× | 43.94 | 25.14 | 26.29 | 71.50 | 91.23 | 42.87 | 62.57 | 57.01 | 52.57 |
| BellowsQ $n$=3 INT8 | 5.42× | 45.62 | 25.74 | 25.55 | 72.50 | 91.53 | 44.06 | 62.03 | 55.67 | 52.84 |
| PolarQuant r2t2v2 | 6.74× | 41.66 | 24.03 | 23.54 | 59.00 | 92.25 | 38.88 | 58.23 | 50.46 | 48.51 |
| BellowsQ $n$=2 INT4 | 6.75× | 46.40 | 24.90 | 26.06 | 72.50 | 91.27 | 44.29 | 61.82 | 56.16 | 52.92 |
| KIVI K2V2 | 7.11× | 42.60 | 24.73 | 26.45 | 71.50 | 91.14 | 43.42 | 61.75 | 55.80 | 52.17 |
| **OTT K2V2** | 7.11× | 44.98 | 24.81 | 26.61 | 72.50 | 91.85 | 43.95 | 64.25 | 57.44 | **53.30** |
| TurboQuant K2V2 | 8.00× | 40.50 | 23.17 | 24.63 | 62.00 | 86.96 | 42.82 | 48.80 | 42.57 | 46.43 |
| RotateKV 2-bit | 8.00× | 37.92 | 23.41 | 26.38 | 61.00 | 87.64 | 39.71 | 53.47 | 46.50 | 47.00 |
| BellowsQ $n$=2 INT3 | 8.53× | 45.89 | 25.30 | 25.80 | 72.50 | 91.60 | 44.29 | 60.80 | 55.62 | 52.72 |
| **BellowsQ $n$=4 INT4** | 12.73× | 45.65 | 24.86 | 25.63 | 72.50 | 91.18 | 44.66 | 60.47 | 54.62 | **52.45** |
| BellowsQ $n$=4 INT3 | 15.85× | 44.39 | 25.09 | 25.57 | 72.50 | 91.38 | 44.03 | 61.01 | 55.40 | 52.42 |
| BellowsQ $n$=4 K3V2 | 18.07× | 44.13 | 25.18 | 25.60 | 72.00 | 90.65 | 44.41 | 60.65 | 55.51 | 52.27 |

Table 12: **Protocol A, Mistral-7B.** Absolute LongBench scores, full test sets, every BellowsKV point and every official baseline configuration, grouped by measured compression tier (rules separate tiers; bins in Table 8); within a tier rows are sorted by ratio and the **best** and second Avg are marked. Baselines run each method's official code; PolarQuant runs its Mixed K+V path, and on Qwen it and TurboQuant quantize bias-free keys (Appendix C.1).

| Method | Comp. | Qasper | QMSum | MNews | TREC | TQA | SAMSum | LCC | RB-P | Avg |
|---|---|---|---|---|---|---|---|---|---|---|
| Dense fp16 | 1.00× | 38.15 | 25.57 | 26.43 | 76.00 | 88.59 | 47.46 | 59.46 | 60.56 | **52.78** |
| BellowsS $n$=2 | 1.96× | 38.75 | 25.62 | 26.06 | 76.00 | 88.81 | 46.78 | 60.36 | 61.97 | 53.04 |
| BellowsS $n$=3 | 2.91× | 38.20 | 25.04 | 25.75 | 76.00 | 89.57 | 47.17 | 59.84 | 61.41 | 52.87 |
| BellowsQ $n$=2 INT8 | 3.69× | 38.49 | 25.34 | 25.81 | 76.00 | 88.56 | 46.74 | 60.42 | 61.99 | 52.92 |
| OTT K4V4 | 3.76× | 37.95 | 25.80 | 26.50 | 76.00 | 88.89 | 47.08 | 60.40 | 61.30 | 52.99 |
| KIVI K4V4 | 3.76× | 36.88 | 25.42 | 26.58 | 76.00 | 88.89 | 47.09 | 60.53 | 60.99 | 52.80 |
| **BellowsS $n$=4** | 3.86× | 37.67 | 25.42 | 25.17 | 76.00 | 90.07 | 46.47 | 60.60 | 62.92 | **53.04** |
| **RotateKV 4-bit** | 4.00× | 39.62 | 24.91 | 26.48 | 77.50 | 88.31 | 46.94 | 59.88 | 61.49 | **53.14** |
| TurboQuant K4V4 | 4.00× | 37.77 | 25.59 | 25.73 | 76.00 | 88.23 | 46.25 | 58.69 | 61.29 | 52.44 |
| RotateKV 3-bit | 5.33× | 34.46 | 25.75 | 26.63 | 76.00 | 87.90 | 45.93 | 59.98 | 61.96 | 52.33 |
| TurboQuant K4V2 | 5.33× | 35.64 | 24.59 | 26.14 | 76.50 | 89.53 | 46.36 | 56.38 | 59.61 | 51.84 |
| BellowsQ $n$=3 INT8 | 5.46× | 39.24 | 24.92 | 25.63 | 76.00 | 89.57 | 47.40 | 59.98 | 61.60 | 53.04 |
| **BellowsQ $n$=2 INT4** | 6.81× | 37.45 | 25.56 | 25.65 | 76.00 | 88.56 | 46.84 | 59.73 | 62.26 | **52.76** |
| KIVI K2V2 | 7.11× | 36.94 | 24.43 | 25.86 | 76.00 | 89.27 | 46.83 | 58.75 | 58.96 | 52.13 |
| OTT K2V2 | 7.11× | 36.55 | 24.66 | 26.36 | 76.00 | 89.47 | 47.03 | 60.25 | 59.79 | 52.51 |
| RotateKV 2-bit | 8.00× | 31.06 | 24.24 | 26.04 | 72.50 | 85.35 | 42.75 | 55.79 | 54.37 | 49.01 |
| TurboQuant K2V2 | 8.00× | 34.83 | 23.12 | 25.06 | 71.50 | 85.96 | 44.64 | 50.24 | 51.13 | 48.31 |
| BellowsQ $n$=2 INT3 | 8.63× | 39.19 | 25.30 | 26.05 | 76.00 | 88.72 | 45.93 | 59.22 | 60.99 | 52.67 |
| BellowsQ $n$=4 INT4 | 12.98× | 38.27 | 25.56 | 24.81 | 75.50 | 89.57 | 46.67 | 60.09 | 62.13 | 52.83 |
| BellowsQ $n$=4 INT3 | 16.25× | 35.95 | 25.49 | 25.34 | 76.00 | 90.32 | 47.03 | 60.03 | 61.88 | 52.75 |
| **BellowsQ $n$=4 K3V2** | 18.59× | 37.83 | 25.23 | 24.55 | 76.00 | 89.82 | 46.88 | 60.47 | 62.21 | **52.87** |

Table 13: **Protocol A, Qwen2.5-7B.** Absolute LongBench scores, full test sets, every BellowsKV point and every official baseline configuration, grouped by measured compression tier (rules separate tiers; bins in Table 8); within a tier rows are sorted by ratio and the **best** and second Avg are marked. Baselines run each method's official code; PolarQuant runs its Mixed K+V path, and on Qwen it and TurboQuant quantize bias-free keys (Appendix C.1).

| Method | Comp. | Qasper | QMSum | MNews | TREC | TQA | SAMSum | LCC | RB-P | Avg |
|---|---|---|---|---|---|---|---|---|---|---|
| Dense fp16 | 1.00× | 43.52 | 23.61 | 23.98 | 71.50 | 89.64 | 45.98 | 60.77 | 66.82 | **53.23** |
| BellowsS $n$=2 | 1.92× | 42.82 | 24.16 | 23.68 | 71.00 | 88.67 | 45.05 | 56.66 | 63.01 | 51.88 |
| BellowsS $n$=3 | 2.86× | 42.99 | 23.50 | 23.48 | 71.00 | 89.00 | 45.12 | 58.00 | 62.25 | 51.92 |
| BellowsQ $n$=2 INT8 | 3.57× | 43.22 | 24.06 | 23.58 | 71.00 | 88.60 | 44.91 | 57.27 | 63.34 | 52.00 |
| **PolarQuant r4t4v4** | 3.66× | 43.94 | 24.42 | 25.52 | 72.00 | 89.46 | 45.76 | 63.68 | 66.41 | **53.90** |
| BellowsS $n$=4 | 3.78× | 42.20 | 23.80 | 22.94 | 71.00 | 88.15 | 44.65 | 55.44 | 61.51 | 51.21 |
| TurboQuant K4V4 | 4.00× | 42.65 | 23.53 | 23.89 | 72.00 | 88.38 | 45.72 | 61.34 | 66.33 | 52.98 |
| **PolarQuant r3t3v4** | 4.13× | 44.13 | 24.02 | 25.54 | 72.00 | 90.10 | 46.49 | 64.07 | 64.75 | **53.89** |
| PolarQuant r2t2v4 | 4.74× | 38.51 | 22.70 | 24.87 | 69.00 | 89.54 | 44.25 | 60.61 | 57.45 | 50.87 |
| BellowsQ $n$=3 INT8 | 5.28× | 43.01 | 23.87 | 23.43 | 71.00 | 88.34 | 45.47 | 58.31 | 61.75 | 51.90 |
| TurboQuant K4V2 | 5.33× | 42.56 | 23.27 | 23.53 | 72.00 | 88.29 | 45.30 | 55.14 | 59.88 | 51.25 |
| BellowsQ $n$=2 INT4 | 6.41× | 42.05 | 24.13 | 23.48 | 71.50 | 88.91 | 45.48 | 56.31 | 63.05 | 51.86 |
| PolarQuant r2t2v2 | 6.74× | 37.63 | 23.09 | 25.41 | 67.50 | 88.08 | 44.44 | 58.11 | 56.74 | 50.12 |
| TurboQuant K2V2 | 8.00× | 36.18 | 21.85 | 22.25 | 66.50 | 81.39 | 42.92 | 55.44 | 50.93 | 47.18 |
| **BellowsQ $n$=2 INT3** | 8.01× | 41.80 | 23.37 | 23.59 | 70.50 | 89.26 | 44.97 | 57.24 | 62.09 | **51.60** |
| **BellowsQ $n$=4 INT4** | 12.15× | 41.84 | 23.54 | 22.60 | 71.00 | 88.60 | 45.45 | 55.98 | 59.98 | **51.12** |
| BellowsQ $n$=4 INT3 | 14.96× | 41.21 | 23.85 | 22.17 | 70.50 | 87.21 | 44.76 | 53.26 | 57.18 | 50.02 |
| BellowsQ $n$=4 K3V2 | 16.92× | 40.13 | 24.44 | 23.84 | 71.00 | 86.45 | 45.74 | 53.27 | 57.40 | 50.28 |

### C.3 PROTOCOL C: FULL LEADERBOARDS

Table 14: **Protocol C Full Table** Single-turn retrieve-then-read on BrowseComp-Plus: the benchmark's oracle prompt filled to 32,768 tokens with its own mined hard negatives, all 830 queries, greedy, the identical context for every method so dense is the ceiling. Score is the benchmark's own judge and Comp. is measured compression with all metadata counted.

| Method | Comp. | Score |
|---|---|---|
| *Qwen3-32B* | | |
| Dense fp16 | 1.00× | 85.54 |
| PolarQuant† | 3.66× | 83.49 |
| KIVI† | 3.76× | 84.94 |
| xKV† | 4.00× | 78.07 |
| PolarQuant† | 6.74× | 51.93 |
| OTT† | 7.11× | 77.95 |
| KIVI† | 7.11× | 75.42 |
| **BellowsQ** | 7.42× | **84.22** |
| CommonKV† | 7.96× | 81.08 |
| xKV† | 8.00× | 26.51 |
| SPECTRA | 9.31× | 81.45 |
| SPECTRA | 11.90× | 78.92 |
| **BellowsQ** | 14.61× | **86.02** |
| CommonKV† | 15.91× | 66.02 |
| xKV† | 16.00× | 8.43 |
| *Qwen3-30B-A3B-Instruct-2507* | | |
| Dense fp16 | 1.00× | 86.14 |
| xKV† | 2.00× | 82.77 |
| PolarQuant† | 3.66× | 80.84 |
| xKV† | 4.00× | 15.54 |
| SPECTRA | 5.36× | 85.66 |
| SPECTRA | 5.58× | 83.13 |
| PolarQuant† | 6.74× | 4.22 |
| OTT† | 7.11× | 81.45 |
| KIVI† | 7.11× | 77.59 |
| **BellowsQ** | 7.42× | **85.30** |
| CommonKV† | 7.96× | 49.88 |
| xKV† | 8.00× | 5.30 |
| **BellowsQ** | 14.61× | **85.78** |

### C.4 THE EVICTION GRID IN FULL

Table 15 is the complete grid behind Section 4.3, all three models and every budget tier from ∼2× to ∼57×, each cell with its measured compression. Rows in one tier are the equal-memory pairs; no pair is byte-exact, BellowsE stores 1–5% more than the control at the paired points, and the margins are several times that gap. The ∼2× and ∼4× tiers hold only the control (BellowsE's shallowest point is already ∼8×), and the same-keep-ratio comparison, where the control wins by 0.8–3.3 points while storing 3.8× more, is read across tiers: the BellowsE row of one tier against the control row two tiers below. Table 16 prices the two currencies directly: from either keep ratio, pruning four times harder buys ∼4× for 2.6–7.8 points, while sharing across four layers buys ∼3.9× for 0.8–1.7 points.

Table 15: **The eviction axis by budget tier (Protocol B, full sample; retention %).** Every route runs the identical verdict; rows are grouped by measured compression so that each tier compares like with like: the per-layer control prunes its own fp16 cache, BellowsE prunes the shared latent, and at ∼30× the composed point meets the naive stack and deep eviction. Comp. spans the three models (Qwen's KV width differs). Best per tier in bold.

| Budget | Route | Keep | Comp. | Llama | Mistral | Qwen |
|---|---|---|---|---|---|---|
| ∼2× | Per-layer eviction | 1/2 | 2.0 | 98.4 | 99.4 | 100.9 |
| ∼4× | Per-layer eviction | 1/4 | 4.0 | 97.3 | 97.7 | 99.5 |
| ∼8× | BellowsE | 1/2 | 7.7–7.8 | **97.6** | **97.9** | **99.2** |
| | Per-layer eviction | 1/8 | 8.0 | 94.7 | 94.9 | 98.3 |
| ∼16× | BellowsE | 1/4 | 15.2–15.4 | **96.1** | **96.4** | **97.9** |
| | Per-layer eviction | 1/16 | 16.0 | 89.5 | 90.7 | 92.6 |
| ∼30× | BellowsQ×BellowsE | 1/2 | 26.3–27.7 | **97.5** | 97.9 | **99.2** |
| | BellowsE | 1/8 | 29.6–30.0 | 92.5 | 93.8 | 95.0 |
| | xKV+INT4+pooled SnapKV | 1/2 | 30.1 | 96.1 | **98.4** | 23.8 |
| ∼57× | BellowsE | 1/16 | 56.6–57.4 | 87.3 | 87.6 | 90.1 |

Table 16: **The exchange rate between the two currencies.** Retention points paid to buy ∼4× by pruning four times harder, against points paid to buy ∼3.9× by sharing across four layers at a fixed keep ratio.

| From keep | Model | prune harder | share |
|---|---|---|---|
| 1/2 | Llama-3.1-8B | 3.74 | 0.82 |
| | Mistral-7B | 4.47 | 1.45 |
| | Qwen2.5-7B | 2.63 | 1.74 |
| 1/4 | Llama-3.1-8B | 7.80 | 1.24 |
| | Mistral-7B | 7.02 | 1.28 |
| | Qwen2.5-7B | 6.87 | 1.57 |

Table 17 is the ∼30× class of Section 4.3 on all three models: the composed operating point against the naive stack and the deep-eviction arm.

Table 17: **The ∼30× budget class, all three models** (Protocol B, full sample; retention %). The composed point stores slightly more than the stack's nominal ratio (27.7/26.3× against 30.1×, a 9–13% byte gap in its favour); the stack's verdict is pooled per group because per-layer verdicts cannot share a basis. Setup in Appendix C.1.

| Route | Comp. | Llama | Mistral | Qwen |
|---|---|---|---|---|
| BellowsQ×BellowsE (evict 1/2) | 27.7/26.3 | **97.5** | 97.9 | **99.2** |
| naive stack: xKV+INT4+pooled SnapKV | 30.1 | 96.1 | **98.4** | 23.8 |
| BellowsE alone (fp16, evict 1/8) | 30.0/29.6 | 92.5 | 93.8 | 95.0 |

### C.5 Equal bit-width on the quantization axis

The tiered tables of Appendix C.2 are the equal-memory comparison: rows in one tier store the same bytes to within the tier's width. Table 18 asks the complementary question by holding the bit-width fixed and varying only what is quantized, each layer's own cache or one latent shared by $n$ layers, which prices the sharing factor itself at equal bits.

Table 18: **Equal bit-width (Protocol A, full sample).** At a fixed bit-width the only difference is what is quantized: each layer's own cache, or one latent shared by $n$ layers. LongBench average and measured compression. Qwen's per-layer baselines are PolarQuant and TurboQuant, both with bias-free keys (Appendix C.1).

| Bits | Method | Comp. | Avg | vs best baseline |
|---|---|---|---|---|
| | *Llama-3.1-8B-Instruct (dense 53.29)* | | | |
| INT4 | KIVI | 3.76 | 53.92 | – |
| | OTT | 3.76 | 53.83 | – |
| | PolarQuant | 3.66 | 53.66 | – |
| | TurboQuant | 4.00 | 52.84 | – |
| | BellowsQ $n$=2 | 6.75 | 52.92 | -1.00 at 1.8× its compression |
| | BellowsQ $n$=4 | 12.73 | 52.45 | -1.47 at 3.4× its compression |
| INT3 | RotateKV | 5.33 | 52.38 | – |
| | BellowsQ $n$=2 | 8.53 | 52.72 | +0.34 at 1.6× its compression |
| | BellowsQ $n$=4 | 15.85 | 52.42 | +0.04 at 3.0× its compression |
| | *Mistral-7B-Instruct-v0.3 (dense 52.78)* | | | |
| INT4 | RotateKV | 4.00 | 53.14 | – |
| | OTT | 3.76 | 52.99 | – |
| | KIVI | 3.76 | 52.80 | – |
| | TurboQuant | 4.00 | 52.44 | – |
| | BellowsQ $n$=2 | 6.81 | 52.76 | -0.38 at 1.7× its compression |
| | BellowsQ $n$=4 | 12.98 | 52.83 | -0.31 at 3.2× its compression |
| INT3 | RotateKV | 5.33 | 52.33 | – |
| | BellowsQ $n$=2 | 8.63 | 52.67 | +0.34 at 1.6× its compression |
| | BellowsQ $n$=4 | 16.25 | 52.75 | +0.42 at 3.0× its compression |
| | *Qwen2.5-7B-Instruct (dense 53.23)* | | | |
| INT4 | PolarQuant | 3.66 | 53.90 | – |
| | TurboQuant | 4.00 | 52.98 | – |
| | BellowsQ $n$=2 | 6.41 | 51.86 | -2.04 at 1.8× its compression |
| | BellowsQ $n$=4 | 12.15 | 51.12 | -2.78 at 3.3× its compression |
| INT3 | BellowsQ $n$=2 | 8.01 | 51.60 | no valid baseline at this bit-width |
| | BellowsQ $n$=4 | 14.96 | 50.02 | no valid baseline at this bit-width |

### C.6 Adaptivity and whisper, per task

Table 3 in the main text holds the rank ladder and the transplant of Section 4.2, Table 19 the transplant per task on all three models, and Figure 6 draws xKV's rank ladder against the rank kept. Table 20 holds the adaptivity and whisper rows behind Section 4.2 per task. The whisper reading is in Appendix B.6: its average effect is small ($\pm 0.1$ at the high tier), its value is the worst case (the 2WikiMQA column), and the V-whisper row's gain at 32k flips sign on short-prefill inputs, which is why it is not the method.

Table 19: **The transplant, per task.** Protocol B retention (%) of xKV at 16× (fp16, rank $p/4$), xKV-hold (the same per-sequence basis at near-full rank with our INT4 leg, 15.1×) and BellowsQ $n$=4 INT4, all three models. The recovery lands on the tasks truncation had cost; on Qwen the arm collapses on every task.

| Arm | Comp. | Qsp | MFQ | HQA | MSQ | NQA | Gov | QMS | PsR | LCC | RBP | 2Wk | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Llama-3.1-8B-Instruct** | | | | | | | | | | | | | |
| xKV (fp16, rank $p/4$) | 16.00× | 91.4 | 80.9 | 82.4 | 87.3 | 73.3 | 75.6 | 94.4 | 95.5 | 92.8 | 86.8 | 58.5 | 83.5 |
| xKV-hold | 15.06× | 97.5 | 95.4 | 96.0 | 100.9 | 96.5 | 96.4 | 99.2 | 99.5 | 101.6 | 97.4 | 92.9 | 97.6 |
| BellowsQ $n$=4 INT4 | 13.84× | 94.6 | 99.0 | 99.1 | 96.4 | 96.2 | 93.5 | 98.7 | 100.0 | 99.7 | 99.0 | 101.5 | 98.0 |
| **Mistral-7B-Instruct-v0.3** | | | | | | | | | | | | | |
| xKV (fp16, rank $p/4$) | 16.00× | 90.4 | 88.5 | 47.8 | 42.1 | 85.2 | 93.0 | 91.8 | 81.7 | 92.7 | 90.0 | 71.1 | 79.5 |
| xKV-hold | 15.06× | 99.8 | 103.0 | 99.9 | 101.5 | 93.7 | 99.4 | 96.9 | 94.9 | 99.8 | 98.9 | 102.9 | 99.2 |
| BellowsQ $n$=4 INT4 | 13.84× | 93.9 | 103.1 | 99.1 | 97.2 | 96.3 | 93.6 | 99.8 | 97.4 | 98.6 | 101.0 | 99.3 | 98.1 |
| **Qwen2.5-7B-Instruct** | | | | | | | | | | | | | |
| xKV (fp16, rank $p/4$) | 16.00× | 63.0 | 69.6 | 26.1 | 22.2 | 38.2 | 80.8 | 78.3 | 48.5 | 74.7 | 61.5 | 35.9 | 54.4 |
| xKV-hold | 15.06× | 17.3 | 16.5 | 26.9 | 28.3 | 22.3 | 10.2 | 31.6 | 30.2 | 21.1 | 29.2 | 24.1 | 23.4 |
| BellowsQ $n$=4 INT4 | 13.13× | 100.8 | 97.3 | 98.6 | 99.2 | 96.8 | 93.6 | 99.9 | 99.0 | 92.6 | 94.5 | 98.6 | 97.3 |

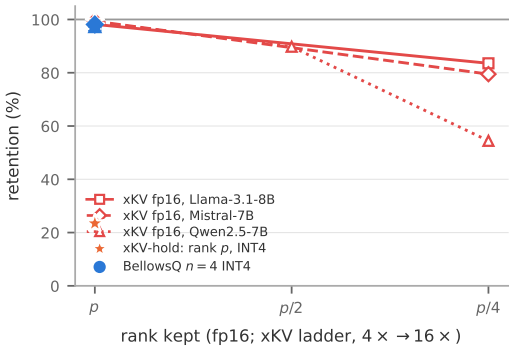



Figure 6: **The rank cliff on the rival's own ladder.** Protocol-B retention of xKV's fp16 basis against the rank kept ($p$, $p/2$, $p/4$, i.e. 4×, 8×, 16×), three models, with xKV-hold (the same basis at rank $p$ stored at INT4, 15.1×) and BellowsQ $n$=4 INT4 at full rank. Cutting rank falls; cutting bits at full rank does not, except on Qwen, where every arm that stores xKV's basis at four bits collapses (Section 4.2).

Table 20: **Adapting to the sequence buys nothing the frozen consensus lacks, and the whisper's value is the worst case, not the mean.** Adaptivity and whisper ablation, per task (Llama-3.1-8B, Protocol B, full sample), the rows behind Section 4.2; the oracle re-decides the basis per sequence at 2.2× prefill cost. Setup in Appendix C.1.

| Variant | Comp. | Qsp | MFQ | HQA | MSQ | NQA | Gov | QMS | PsR | LCC | RBP | 2Wk | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BellowsQ (final) | 13.84 | 94.6 | 99.0 | 99.1 | 96.4 | 96.2 | 93.5 | 98.7 | 100.0 | 99.7 | 99.0 | 101.5 | 98.0 |
| + V-whisper | 13.13 | 94.9 | 99.1 | 98.2 | 98.5 | 99.3 | 95.2 | 100.7 | 99.5 | 101.0 | 98.9 | 100.7 | 98.7 |
| − whisper | 13.90 | 93.3 | 97.5 | 99.0 | 97.1 | 97.5 | 95.4 | 100.4 | 100.0 | 100.7 | 98.6 | 97.5 | 97.9 |
| + per-sequence gauge | 13.90 | 93.2 | 94.8 | 99.5 | 98.5 | 93.5 | 92.3 | 98.6 | 99.0 | 99.1 | 97.8 | 97.8 | 96.7 |
| per-seq SVD oracle | 15.06 | 91.6 | 92.8 | 98.8 | 99.5 | 95.3 | 99.9 | 99.5 | 99.5 | 96.3 | 96.4 | 101.2 | 97.3 |
| xKV-hold (+ our INT4) | 15.06 | 97.5 | 95.4 | 96.0 | 100.9 | 96.5 | 96.4 | 99.2 | 99.5 | 101.6 | 97.4 | 92.9 | 97.6 |
| 8× point ($n$=2, INT4) | 6.90 | 96.3 | 97.7 | 100.3 | 97.6 | 98.6 | 98.6 | 101.1 | 100.0 | 99.6 | 99.2 | 101.6 | 99.2 |
| 12× point ($n$=3, INT4) | 10.40 | 93.8 | 96.7 | 99.1 | 102.5 | 97.2 | 97.0 | 99.9 | 100.0 | 99.4 | 98.9 | 100.3 | 98.6 |