PUSHING THE KV CACHE BEYOND THE 2-BIT CLIFF VIA SPECTRAL TRANSFORM CODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are increasingly applied to very long inputs, such as whole documents, codebases, and conversations that run for many turns. The key-value (KV) cache stores the attention keys and values of all past tokens. Because the cache grows with the context length and is re-read in full at every generated token, it dominates the inference memory at long context. Most existing methods reduce this cost by quantizing the cache: storing every value at the same low precision. These methods reach about two bits per value, but they rarely go below two bits, because accuracy drops sharply at that point. The reason is that two bits allow only four levels per value, which are too few for a cache with outlier channels, which is known as the 2-bit cliff. In our study, we find that once we project the cache into a latent space that removes the correlations between channels, the contributions of the different channels of the KV cache to the attention output are highly uneven. And we find that spending the bit budget on those few important channels is far more accurate than spreading the bit budget evenly. Guided by these observations, we develop SPECTRA, a training-free compression method for the KV cache that requires no change to the model. SPECTRA transforms the KV cache into a latent space in which the channels are ordered by their contribution to the attention output, and then allocates the bit budget across these ordered channels. On long-context benchmarks and three instruction-tuned models, every uniform quantization methods that we compare against has fallen below the 2-bit cliff at 8× compression rate, and none of these methods has a configuration beyond 9×. SPECTRA matches the uncompressed model at 8× on Llama-3.1-8B, stays within 1 point of that accuracy up to 11×, and recovers every planted needle in needle-in-a-haystack at 12×.

1 INTRODUCTION

Large language models (LLMs) now power a growing range of agentic applications. A coding assistant may work over an entire repository, and a dialogue agent may run for many turns while calling external tools Yao et al. (2023); Schick et al. (2023); Jimenez et al. (2024); Yang et al. (2024). What these applications have in common is that the length of the context, together with the model's own generated reasoning, can reach hundreds of thousands of tokens Team et al. (2024); Hsieh et al. (2024); Bai et al. (2024b). At this scale the main cost of LLM serving gradually shift from model weights to the key-value (KV) cache, which stores the attention keys and values of past tokens so that the model does not recompute them during decoding. Because every new token attends to the whole cache, the cache must sit in fast memory and be reloaded at each step, and then its size grows with both the context length and the number of concurrent requests. As a result, at long context the KV cache can grow larger than the model weights themselves Kwon et al. (2023); Zheng et al. (2024), and thus it becomes the main constraint on the context length and the number of concurrent requests that the serving hardware can support.

Many methods compress the KV cache, and they reduce it along one of three axes. Token eviction methods reduce the cache by discarding tokens they judge unimportant, such as H2O Zhang et al. (2023) and SnapKV Li et al. (2024), which keep the tokens that have received the most attention, or StreamingLLM Xiao et al. (2024b), which keeps the first and the most recent tokens. Low-rank methods reduce the cache by storing keys and values in fewer dimensions, such as Palu Chang et al. (2024) and Eigen Attention Saxena et al. (2024), which project them onto a low-rank subspace

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.

obtained from the model's weights or activations. Quantization methods reduce the cache by storing every value at lower precision, such as KIVI Liu et al. (2024b) and KVQuant Hooper et al. (2024), which quantize keys per channel and values per token to contain outlier channels. These families differ in what they remove, yet most share the same strategy: they fix the budget in advance, as a set of tokens, a target rank, or a bit-width, and rarely adapt it to how much each part of the cache matters for the output. This limitation is clearest for quantization. Lowering every value to the same precision reaches about two bits, but rarely goes further, because at this so-called 2-bit cliff Dettmers & Zettlemoyer (2023); Zhou et al. (2026), accuracy drops sharply: with only four levels per value, a few large entries take up the range and the many small ones collapse into noise Xiao et al. (2024a); Dettmers et al. (2022). Appendix A discusses these families and their relation to our method in full.

In this paper, we take a different point of view, that rather than fixing any of these choices in advance, we ask where each bit of the budget should go, and treat KV-cache compression as one problem: how to spend a limited budget on the parts of the cache that matter most. From this point of view, the three axes we mentioned in the previous paragraph are settings of a single allocation: low-rank projection removes channels, while quantization sets how many bits each channel keeps, and a removed channel is just a channel kept at zero bits. Choosing a rank and choosing a precision are therefore the same decision: how many bits to give each channel, from several down to zero. To make this decision well, we first need to know which channels matter, yet the raw cache does not reveal this, because its channels are correlated and none stands out on its own. We find that the answer appears once the cache is rotated into its own principal directions, the orthogonal axes along which it varies the most, which we estimate from its own statistics. In this basis the channels are uncorrelated, and a small set of them can represent for most of the cache.

Building on this, we introduce SPECTRA, a training-free and drop-in method that rotates the cache into its spectral basis, the principal directions along which it varies, and spends its bit budget on the channels that carry the signal. SPECTRA needs only a small, one-time calibration to estimate this basis, after which its channels are uncorrelated and ordered by how much they contribute. At inference, it stores each key and value in this basis and gives every channel a bit-width that matches its contribution: more bits to the few dominant channels, and fewer, down to zero, for the rest. A channel given zero bits is simply removed, so the single rule that sets each channel's precision also sets its rank. Because the basis is a small fixed matrix that folds into attention, SPECTRA reconstructs the cache with little overhead and without fine-tuning. Our contributions are summarized as follows:

- **An analysis of the KV cache showing that its information is concentrated in a few of its principal directions, where a small set of channels carries most of the signal.** This recasts KV-cache compression as spending a fixed bit budget on the channels that matter, and unifies low-rank projection and quantization as the two ends of one control, since a channel given zero bits is simply dropped.
- **SPECTRA, a training-free and drop-in KV-cache method that transforms the cache into these principal directions and allocates its bits to the channels that carry the signal.** We evaluate SPECTRA on Llama-3.1-8B, Mistral-7B, and Qwen2.5-7B across LongBench and needle-in-a-haystack retrieval, under one harness and on the full test split of every task. Where uniform quantization still works, below about five times compression, SPECTRA is level with the strongest of it. Past eight times, where uniform quantization breaks down, SPECTRA is the only method with operating points at all, and on Llama-3.1-8B it stays within a point of the dense model out to eleven times.

2 Motivation: Where Should Each Bit Go?

We view the KV cache as a signal to be coded rather than a tensor to be compressed, so as a result, compressing it is then a rate–distortion problem, in which the question is not how many bits to use or which tokens to keep, but where each bit should be allocated. This reframes KV-cache compression from an engineering heuristic into a coding problem, and raises a concrete question: given a fixed budget, in which basis and by what measure of importance should each bit be allocated across channels and precision? We answer it by probing the KV cache of pretrained LLMs directly, scoring the importance each choice by the error it induces in the attention output on held-out text. The statistics that define our transform come from a small calibration set. **The three observations below answer this question in turn.**

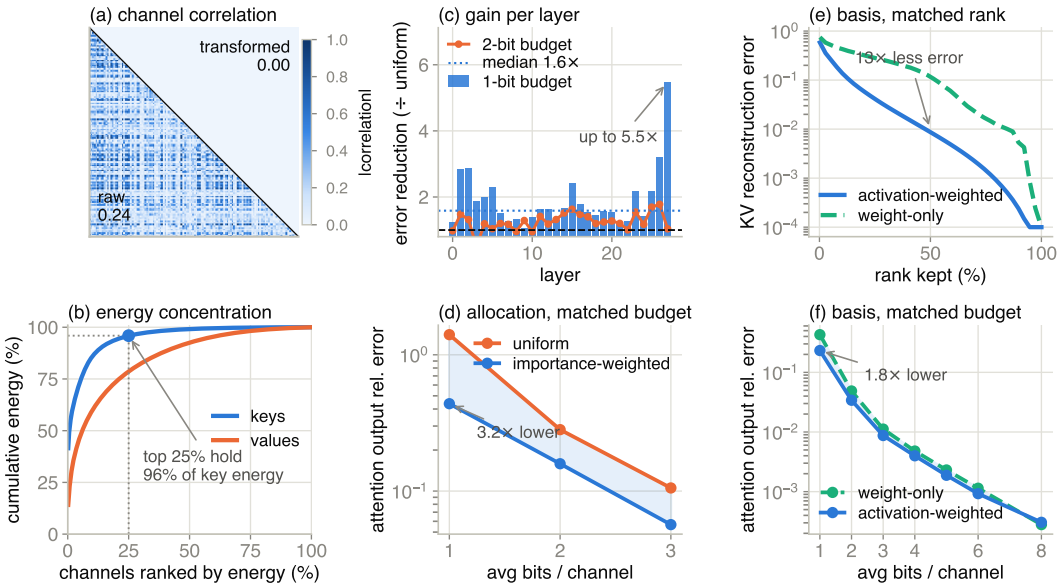

Figure 1: **The three observations that fix where each bit should be allocated** (Qwen2.5-7B, and the other models behave the same way). *OB1, left column.* **(a)** Magnitude of the key channel-correlation matrix, raw channels in the lower triangle and the same channels after rotating into a basis computed from their own statistics in the upper, with the mean off-diagonal printed in each. **(b)** Cumulative energy over channels ranked by energy. *OB2, middle column.* **(c)** Per-layer reduction in attention-output error from importance-weighted allocation over uniform, at a one-bit and a two-bit budget. **(d)** The same error against average bits per channel at a representative layer. *OB3, right column.* **(e)** KV reconstruction error against the fraction of rank kept, and **(f)** attention-output error against average bits per channel, both comparing a transform built from the activation statistics $\mathbf{G} = \mathbb{E}[\mathbf{h}^\top \mathbf{h}]$ with a weight-only transform.

OB1: The cache's importance is concentrated, but only in the right basis In the raw channel basis, the keys within a layer are strongly correlated, so no channel stands out as more important than another and there is no natural axis to allocate along (Figure 1a, lower triangle: mean off-diagonal correlation ≈ 0.2). Once we rotate the cache into a basis computed from its own statistics, the channels become uncorrelated (upper triangle: ≈ 0), and their energy concentrates sharply: the top 25% of channels carry 96% of the key energy, and the rest are nearly redundant (Figure 1b). Importance is therefore separable and heavily skewed, but only after this transform, and this tells us to allocate the bits in the transformed basis rather than the raw one.

OB2: Allocating bits by importance beats uniform precision, and most sharply below two bits. Because the channels differ so much in energy, giving each the same number of bits wastes the budget. When we instead give more bits to high-energy channels and fewer, even down to zero, to low-energy ones, the attention-output error drops at almost every layer (Figure 1c), by a median of $1.6\times$ and up to $5.5\times$ at one bit per channel. The advantage is concentrated where uniform quantization fails: the gap is largest at one bit and closes as the bit-width grows (Figure 1d), exactly the sub-two-bit regime that gives the 2-bit cliff its name. A channel assigned zero bits is simply dropped, so this single allocation spans both quantization and rank reduction, and this make low-rank projection and quantization the two ends of one control.

OB3: A channel's importance is set by the data, not its weights. We observe that how much a channel matters depends on how strongly the input activations drive it, not on the magnitude of its projection weights. When we build the transform and rank the channels from the activation statistics $\mathbf{G} = \mathbb{E}[\mathbf{h}^\top \mathbf{h}]$, where $\mathbf{h}$ is the layer input to the key/value projection, rather than from the weight matrix alone, we keep far more of the cache at the same budget: at half the rank the reconstruction error is about $13\times$ lower (Figure 1e), and at a matched bit-rate the attention-output error is about $1.9\times$ lower (Figure 1f).

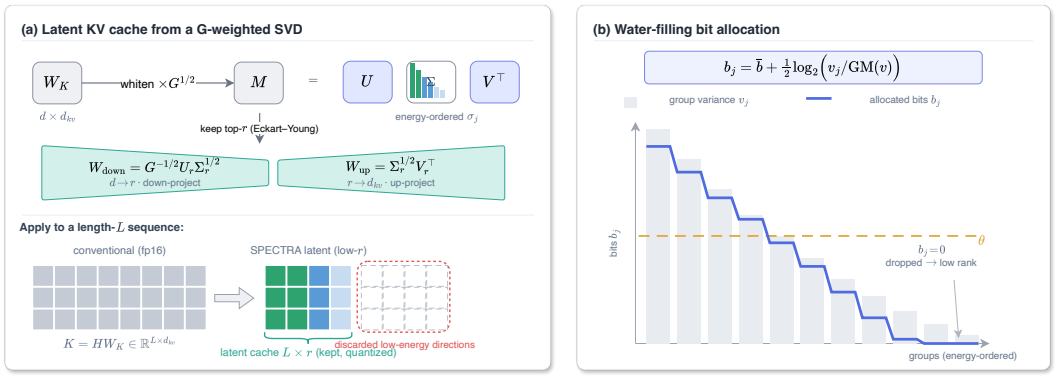

Figure 2: **SPECTRA method overview. (a) Latent KV cache from a G-weighted SVD.** Each key/value projection is factorized from the whitened weight $M = G^{1/2}W$, and keeping the top-r singular directions (Eckart–Young) yields $W_{\text{down}}, W_{\text{up}}$, so a length-$L$ sequence is cached as an $L \times r$ energy-ordered latent instead of the $L \times d_{kv}$ full-precision K, V, and the low-energy directions are discarded by the truncation. **(b) Water-filling bit allocation.** Reverse water-filling against a level θ assigns more bits to high-variance latent groups and drives the low-energy tail to zero bits, which is rank reduction by another name. At runtime only the latent is stored, and K, V are reconstructed on the fly.

Together these observations specify our method. We encode the cache in an activation-weighted transform that decorrelates and concentrates it (Observations 1 and 3), and we spend a fixed bit budget across the resulting channels by importance, letting the weakest channels fall to zero bits (Observation 2). Section 3 makes this precise as SPECTRA.

3 Method

The three observations recast KV-cache compression as deciding where to spend a fixed budget, and that leaves one design choice to make concrete: what the cache should store. The answer follows from how redundant the produced keys and values are. Measured in the metric of the model's activations, most of their energy occupies a small subspace, so we cache a compact latent in place of $\mathbf{K}$ and $\mathbf{V}$ and recover them only when they are needed. This single decision organizes the rest of the section. We first build the latent through a G-weighted transform that is second-moment-orthogonal and energy-ordered by construction (Section 3.1). We then quantize that latent at a rate–distortion optimum which the same structure makes well-posed (Section 3.2). Notation follows Appendix B.

3.1 Caching the Latent: A G-weighted Transform

We now make the latent precise. We seek a rank-r factorization of each key projection, $\mathbf{W}_K \approx \mathbf{W}_{\text{down}}\mathbf{W}_{\text{up}}$ with $\mathbf{W}_{\text{down}} \in \mathbb{R}^{d \times r}$ and $\mathbf{W}_{\text{up}} \in \mathbb{R}^{r \times d_{kv}}$ ($r \le d_{kv}$), so that the cache holds the latent $\mathbf{c} = \mathbf{h}\mathbf{W}_{\text{down}}$ and recovers a key as $\hat{\mathbf{k}} = \mathbf{c}\,\mathbf{W}_{\text{up}}$ only when it is needed. Over a length-L sequence this replaces the conventional $L \times d_{kv}$ cache $\mathbf{K} = \mathbf{H}\mathbf{W}_K$ with the $L \times r$ latent $\mathbf{C} = \mathbf{H}\mathbf{W}_{\text{down}}$, reconstructing $\hat{\mathbf{K}} = \mathbf{C}\,\mathbf{W}_{\text{up}}$ on demand (Figure 2a). The directions beyond rank r are discarded by the truncation itself, so the discarded tail is a rank reduction rather than a separate pruning step.

The question is what "$\approx$" should mean here. It should not mean that $\mathbf{W}_K \approx \mathbf{W}_{\text{down}}\mathbf{W}_{\text{up}}$ in the ordinary Frobenius sense, because that treats every direction of the weight as equally important. What we actually care about is that the keys the model *produces* are preserved, $\mathbf{h}\mathbf{W}_K \approx \mathbf{h}\,\mathbf{W}_{\text{down}}\mathbf{W}_{\text{up}}$, on the activations $\mathbf{h}$ that actually occur. The right objective is therefore an expectation over the model's own activation distribution,

$$\min_{\operatorname{rank} r} \; \mathbb{E}_{\mathbf{h}} \left\| \mathbf{h}\left(\mathbf{W}_K - \mathbf{W}_{\text{down}}\mathbf{W}_{\text{up}}\right) \right\|^2, \tag{1}$$

and identically for $\mathbf{W}_V$.

This expectation has a convenient closed form. With $\mathbf{G} = \mathbb{E}_{\mathbf{h}}[\mathbf{h}^\top \mathbf{h}]$ the uncentered second moment of the layer input, a short calculation given in Appendix C turns Equation equation 1 into the G-weighted low-rank problem $\min_{\mathrm{rank}\, r} \|\mathbf{G}^{1/2}(\mathbf{W}_K - \mathbf{W}_{\text{down}}\mathbf{W}_{\text{up}})\|_F$. The factor $\mathbf{G}^{1/2}$ re-weights the reconstruction error by how often each input direction is actually used. A direction the model visits constantly is expensive to get wrong, while one it never visits costs nothing, no matter how large the corresponding weights are. This is the one place SPECTRA departs from plain low-rank factorization of the cache, such as Palu (Chang et al., 2024) or SVD-LLM (Wang et al., 2024). We minimize error in the metric of the activations rather than in raw weight space, and our experiments confirm that this distinction shows up in end-task accuracy.

The G-weighted problem has a closed-form solution. We whiten the weight as $\mathbf{M} = \mathbf{G}^{1/2}\mathbf{W}_K$, so that reconstruction error is measured in the activation metric, and take its SVD $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. Keeping the top r components is optimal by Eckart–Young (Eckart & Young, 1936) in this metric, and undoing the whitening with a *square-root* split of the singular values gives

$$\mathbf{W}_{\text{down}} = \mathbf{G}^{-1/2}\,\mathbf{U}_r\,\mathbf{\Sigma}_r^{1/2}, \qquad \mathbf{W}_{\text{up}} = \mathbf{\Sigma}_r^{1/2}\,\mathbf{V}_r^\top. \tag{2}$$

Any split of $\mathbf{\Sigma}_r$ between the two factors reconstructs equally well. What the square-root split buys is the statistics of the cached quantity itself: it makes the latent's uncentered second moment equal to $\mathbf{\Sigma}_r$, which is diagonal and descending. The latent coordinates are therefore orthogonal in uncentered second moment and ordered by energy, with coordinate j carrying second moment $(\mathbf{\Sigma}_r)_{jj}$. That canonical ordering is what separates the transform from a plain SVD, which does not use the activation metric, and from the random rotations of TurboQuant (Zandieh et al., 2025) and QuaRot (Ashkboos et al., 2024), which impose no energy ordering at all. Appendix C gives the identity and the alternative splits.

Two refinements complete the transform. First, the low-rank factorization of the value path leaves a small structured residual at the attention output. Because the output projection $\mathbf{W}_O$ lies outside the cache, we absorb the dominant part of that residual with a closed-form, training-free reduced-rank correction to $\mathbf{W}_O$ (Appendix D), which changes neither the cached object nor the cost of inference. Second, the rank budget B is split asymmetrically between the two paths, as $r_k = \kappa B$ and $r_v = (1-\kappa)B$. The produced energy of keys concentrates faster than that of values, so keys tolerate a smaller rank and values receive the larger share ($\kappa < \frac{1}{2}$). Both factors and the $\mathbf{W}_O$ correction fold back into the model weights, so the deployed model is a drop-in replacement of $\mathbf{W}_K$ and $\mathbf{W}_V$ with no extra runtime structure.

3.2 Rate–Distortion Quantization of the Latent

Because the latent coordinates are energy-ordered, quantizing them all at one bit-width is wasteful. A high-variance coordinate and a near-zero one would be given the same precision even though they contribute very differently to the reconstruction. We therefore subtract the measured coordinate means and model the centered coordinates as independent Gaussian scalar sources with marginal variances v_j. The independence is an approximation, because the G-weighted transform guarantees a diagonal uncentered second moment rather than a diagonal centered covariance. Under this model, quantizing coordinate j at b_j bits costs distortion proportional to $v_j\, 2^{-2b_j}$ at high rate (Gersho & Gray, 1992; Gray & Neuhoff, 1998), and minimizing the total distortion at a fixed average rate $\bar{b}$ yields the classical *reverse water-filling* allocation (Huang & Schultheiss, 1963; Cover & Thomas, 2006) (Figure 2b),

$$b_j = \bar{b} + \tfrac{1}{2}\log_2\Big(v_j \,/\, \mathrm{GM}(v)\Big), \qquad \mathrm{GM}(v) = \Big(\textstyle\prod_j v_j\Big)^{1/r}, \tag{3}$$

clamped to $[0, b_{\max}]$. Half a bit is added for every doubling of variance above the geometric mean, and the optimum equalizes the distortion left in each retained coordinate rather than the bits spent on it. Appendix E derives the allocation. What the allocation needs operationally is exactly what the transform supplies, a canonical energy ordering and measured marginal variances, and together the two follow the classical transform-coding pattern (Goyal, 2001) of transforming a source into an ordered coordinate system and then allocating rate across its coefficients.

Two properties make the allocation practical (Figure 4, Appendix E). First, a coordinate given zero bits is dropped entirely, so as $\bar{b}$ falls the low-energy tail is zeroed and quantization passes *continuously* into rank reduction. The effective rank is then an output of the allocator rather than a separate

hyperparameter. Second, per-coordinate bit-widths do not compose with a streaming cache, so we group g adjacent energy-ordered coordinates to share one bit-width, taken from Equation equation 3 at the group's mean variance, and quantize each group with per-token min and max scales. This keeps the cache append-only and robust to outlier tokens. Each group stores its payload plus one fp16 scale and zero-point per token, both counted in the effective rate, and zero-bit groups store nothing.

4 EXPERIMENTS

4.1 SETTINGS

Models and tasks. We evaluate SPECTRA on Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) and Qwen2.5-7B-Instruct (Qwen et al., 2025). All three use grouped-query attention. Llama and Mistral have 32 query and 8 key–value heads, giving a 1024-dimensional per-token KV cache, while Qwen2.5-7B has 28 query and 4 key–value heads and therefore half that width, which matters below. **LongBench** (Bai et al., 2024a) contributes eight English tasks spanning single-document QA, summarization, few-shot classification and QA, and repository-level code completion, and **Needle-in-a-Haystack** (NIAH) measures retrieval over six context lengths (1k–32k) and seven depths, with one needle and with four. **Every number we report uses the full test split of each task.** SPECTRA is applied post hoc on Hugging Face Transformers with no fine-tuning, after one calibration pass of 16 windows of 2,048 tokens, half WikiText-2 and half source code, using no evaluation example.

Protocol and baselines. We compare against KIVI (Liu et al., 2024b), TurboQuant (Zandieh et al., 2025), PolarQuant (Han et al., 2025), OTT (Su et al., 2025a) and RotateKV (Su et al., 2025b), each run from its official implementation. Every method, the dense model and SPECTRA included, goes through one identical harness: official LongBench prompts and metrics, contexts capped at $31{,}500$ tokens, and greedy decoding. All compression ratios are effective, counting the metadata each method stores and byte-weighted over the examples actually evaluated. Coverage follows what the released code supports, so Qwen is compared against PolarQuant and TurboQuant only, and two compatibility fixes were needed, one for PolarQuant's key-only default path and one for the biased key projection on Qwen2.5. Everything else, including the calibration ablation and the full sweep, is in Appendices G and I.

4.2 OVERALL RESULTS

Reading a row of Table 1 to the right shows how far that method goes before it runs out of operating points, and that is the result of this paper. Below about $5\times$ the two families are tied and the best quantizer is ahead. It takes the $\sim 4\times$ bin on all three models, by 0.74 on Llama (KIVI at 53.92 against SPECTRA's 53.18), by 0.20 on Mistral and by 0.99 on Qwen. This is the regime in which rounding alone costs almost nothing, so a transform can only add its own reconstruction error on top. In the $\sim 5\times$ bin SPECTRA takes all three models, and around $7\times$ OTT recovers the lead on Llama and Mistral, at 53.30 and 52.51 against 52.88 and 52.30, while storing slightly less.

The picture changes at $8\times$, which is the 2-bit cliff of Section 2. A per-layer quantizer can only reach that bin by rounding every stored value to two bits, and on Llama RotateKV and TurboQuant fall to 47.00 and 46.43 from a dense 53.29. SPECTRA reaches it by spending an uneven budget across the latent instead of a uniform one across raw channels, and scores 53.29 on Llama, the dense average to two decimals, and 52.07 on Mistral. Past that bin every baseline column is empty, while SPECTRA holds 52.99 at $8.82\times$ and 52.44 at $11.09\times$ on Llama, still within a point of dense.

Qwen2.5-7B needs a narrower latent for the same bins because its cache is half as wide, so the widths tuned on the other models stop near $5.6\times$ and four narrower ones cover the rest (Table 3, which is also where the equal-ratio comparison lives). At exactly $8.00\times$, the ratio TurboQuant reaches, SPECTRA scores 48.58 against its 47.18, and at $8.82\times$ it scores 47.67 where no baseline has a point. Qwen is still where the ceiling bites hardest, sitting 4.6 points below dense at $8\times$ against the dense average on Llama.

Please see full result on 3 models paired with LongBench and NIAH task in the Appendix I

Table 1: **Every baseline runs out before $9\times$, and SPECTRA is the only method with operating points beyond it on all three models.** LongBench average over the eight English tasks, full test split, at each method's best operating point in each compression bin, with that point's measured ratio as a subscript and all metadata counted. An empty cell means no operating point in the bin. Best per bin is **highlighted**. KIVI, OTT and RotateKV have no Qwen2.5 path in their released code.

Method	∼4×	∼5×	∼7×	∼8×	∼9×	∼11×
Llama-3.1-8B-Instruct dense (fp16) 53.29						
SPECTRA	$53.18_{3.6}$	**$52.62_{4.6}$**	$52.88_{6.7}$	**$53.29_{7.6}$**	**$52.99_{8.8}$**	**$52.44_{11.1}$**
KIVI	**$53.92_{3.8}$**	–	$52.17_{7.1}$	–	–	–
OTT	$53.83_{3.8}$	–	**$53.30_{7.1}$**	–	–	–
Palu	–	$52.27_{4.4}$	–	$48.00_{7.5}$	–	–
PolarQuant	$53.66_{3.7}$	$50.63_{4.7}$	$48.51_{6.7}$	–	–	–
RotateKV	–	$52.38_{5.3}$	–	$47.00_{8.0}$	–	–
TurboQuant	$52.84_{4.0}$	$52.57_{5.3}$	–	$46.43_{8.0}$	–	–
Mistral-7B-Instruct-v0.3 dense (fp16) 52.78						
SPECTRA	$52.94_{4.0}$	**$52.95_{4.6}$**	$52.30_{6.8}$	**$52.07_{7.7}$**	**$52.32_{8.9}$**	**$50.76_{11.3}$**
KIVI	$52.80_{3.8}$	–	$52.13_{7.1}$	–	–	–
OTT	$52.99_{3.8}$	–	**$52.51_{7.1}$**	–	–	–
Palu	–	$52.77_{4.4}$	–	$50.19_{7.5}$	–	–
RotateKV	**$53.14_{4.0}$**	$52.33_{5.3}$	–	$49.01_{8.0}$	–	–
TurboQuant	$52.44_{4.0}$	$51.84_{5.3}$	–	$48.31_{8.0}$	–	–
Qwen2.5-7B-Instruct dense (fp16) 53.23						
SPECTRA	$52.91_{3.5}$	**$53.54_{4.4}$**	**$50.84_{5.6}$**	**$48.72_{7.7}$**	**$47.67_{8.8}$**	–
Palu	–	$52.06_{4.4}$	–	$48.50_{7.4}$	–	–
PolarQuant	**$53.90_{3.7}$**	$50.87_{4.7}$	$50.12_{6.7}$	–	–	–
TurboQuant	$52.98_{4.0}$	$51.25_{5.3}$	–	$47.18_{8.0}$	–	–

Table 2: **LongBench per task, Llama-3.1-8B-Instruct.** One SPECTRA row per compression tier against all baselines, plus the high-compression points no baseline reaches. Avg is the mean over the eight tasks and every row is the full test split. Within each tier the **best** and second Avg are highlighted.

Method	Comp.	Qasper	QMSum	MNews	TREC	TQA	SAMSum	LCC	RB-P	Avg
base (fp16)	1.00×	46.57	25.19	27.03	72.50	91.65	43.47	63.38	56.55	**53.29**
SPECTRA	3.56×	46.14	25.57	26.90	72.50	92.17	43.65	63.42	55.08	53.18
PolarQuant	3.66×	46.01	25.56	26.69	72.50	91.86	42.88	65.38	58.44	53.66
KIVI	3.76×	46.11	25.14	27.11	72.50	92.42	44.10	65.05	58.91	**53.92**
OTT	3.76×	46.47	25.30	26.66	72.50	91.73	43.73	64.96	59.28	53.83
TurboQuant	4.00×	45.87	25.21	26.79	73.00	91.06	43.68	62.44	54.65	52.84
PolarQuant	4.13×	45.60	25.38	26.60	72.50	92.16	42.44	65.36	57.97	53.50
Palu	4.41×	45.12	24.67	26.63	73.00	92.15	44.73	59.72	52.11	52.27
SPECTRA	4.57×	44.48	25.04	27.02	71.50	90.90	43.43	63.14	55.47	**52.62**
PolarQuant	4.74×	42.52	24.37	25.46	66.00	91.50	40.91	61.13	53.18	50.63
RotateKV	5.33×	45.67	24.69	26.85	73.50	90.34	42.86	61.42	53.71	52.38
TurboQuant	5.33×	43.94	25.14	26.29	71.50	91.23	42.87	62.57	57.01	52.57
SPECTRA	6.74×	44.20	25.51	25.83	70.50	90.14	44.11	61.83	60.95	52.88
PolarQuant	6.74×	41.66	24.03	23.54	59.00	92.25	38.88	58.23	50.46	48.51
KIVI	7.11×	42.60	24.73	26.45	71.50	91.14	43.42	61.75	55.80	52.17
OTT	7.11×	44.98	24.81	26.61	72.50	91.85	43.95	64.25	57.44	**53.30**
Palu	7.51×	36.22	24.07	24.86	69.50	86.60	42.49	49.79	50.49	48.00
SPECTRA	7.65×	45.34	25.12	26.17	71.50	91.78	43.40	61.77	61.26	**53.29**
TurboQuant	8.00×	40.50	23.17	24.63	62.00	86.96	42.82	48.80	42.57	46.43
RotateKV	8.00×	37.92	23.41	26.38	61.00	87.64	39.71	53.47	46.50	47.00
SPECTRA	8.82×	45.84	25.24	25.69	73.00	89.77	43.64	62.77	57.95	**52.99**

Retrieval separates the families more sharply than the averages do (Table 10, grids in Appendix I.2). Single-needle recovery for SPECTRA is 100% at every tier tested on all three models, including $12.2\times$ on Llama, while the 2-bit baselines lose cells at the same compression or less: TurboQuant recovers 88.1% on Llama and 66.7% on Mistral, and PolarQuant only 50.0% on Llama at $4.7\times$. The four-needle variant widens the gap, where SPECTRA at $12.2\times$ recovers every planted needle on Llama, above the dense model's 85.7%, against 63.1% for TurboQuant.

Table 3: **LongBench per task, Qwen2.5-7B-Instruct.** Full sweep: every SPECTRA operating point and all baselines, grouped by compression regime. Avg is the mean over the eight tasks and every row is the full test split. Within each tier the **best** and second Avg are highlighted.

Method	Comp.	Qasper	QMSum	MNews	TREC	TQA	SAMSum	LCC	RB-P	Avg
base (fp16)	1.00×	43.52	23.61	23.98	71.50	89.64	45.98	60.77	66.82	**53.23**
SPECTRA	3.48×	43.26	23.81	24.11	71.50	88.85	45.52	59.35	66.90	52.91
PolarQuant r4t4v4	3.66×	43.94	24.42	25.52	72.00	89.46	45.76	63.68	66.41	**53.90**
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
SPECTRA	4.36×	45.15	23.55	24.30	71.50	88.93	46.65	62.59	65.63	**53.54**
SPECTRA	4.43×	44.66	23.36	23.61	71.50	88.81	44.91	59.92	64.16	52.62
SPECTRA	4.91×	42.32	23.38	24.01	68.00	89.49	45.14	60.69	60.78	51.73
SPECTRA	4.92×	42.96	23.40	23.05	70.00	88.83	46.04	59.76	64.14	52.27
Palu r0.9 4-bit	4.37×	42.55	23.87	24.16	72.50	86.86	45.10	57.65	63.81	52.06
PolarQuant r2t2v4	4.74×	38.51	22.70	24.87	69.00	89.54	44.25	60.61	57.45	50.87
TurboQuant K4V2	5.33×	42.56	23.27	23.53	72.00	88.29	45.30	55.14	59.88	51.25
SPECTRA	5.64×	39.11	23.66	24.09	71.50	86.77	46.26	60.47	54.90	**50.84**
PolarQuant r2t2v2	6.74×	37.63	23.09	25.41	67.50	88.08	44.44	58.11	56.74	50.12
SPECTRA	7.65×	31.01	23.42	25.52	70.00	89.69	45.17	53.04	51.87	**48.72**
SPECTRA	8.00×	32.53	23.16	25.14	72.50	89.48	44.04	50.15	50.74	48.47
SPECTRA	8.00×	31.05	23.04	25.58	70.50	90.11	43.49	53.40	51.49	48.58
Palu r0.7 3-bit	7.37×	32.84	23.95	25.27	73.00	86.61	44.76	51.31	50.24	48.50
TurboQuant K2V2	8.00×	36.18	21.85	22.25	66.50	81.39	42.92	55.44	50.93	47.18
SPECTRA	8.82×	34.18	21.85	25.24	70.00	88.20	43.29	52.13	46.50	**47.67**

4.3 WHY CACHE A TRANSFORMED LATENT?

We cache a *transformed* latent rather than the raw cache because the transform concentrates the produced KV energy into a few ordered directions (Section 2), and that concentration is what lets SPECTRA spend its budget on two axes at once. Figure 3 makes the consequence concrete. Holding compression fixed and sweeping the latent width, the 2-bit curve is not merely worse than the others, it is unusable, scoring 15.84 at 7.11× where the 4-bit curve still scores 52.83 at 7.53×. Since rank reduction and quantization are the two ends of one control (Section 3), SPECTRA moves along the upper envelope of these curves instead, trading rank for precision as the target tightens. The two upper curves meet near 10×, where 3 bits at a wider latent (49.30 at 9.85×) draws level with 4 bits at a narrower one (48.88 at 10.04×).

Table 4: **SPECTRA decodes at 2–3.4× dense throughput from the same GPU memory.** Llama-3.1-8B-Instruct on one H100 NVL under HuggingFace: the dense fp16 cache at the largest batch the memory budget holds (64 / 16 / 4 sequences at 8k / 32k / 128k)

variant	ratio	throughput vs. dense at equal memory (batch)		
		8k	32k	128k
SPECTRA R_K=384	8.6×	2.65× (522)	3.15× (135)	3.43× (34)
SPECTRA R_K=512	7.5×	2.36× (460)	2.77× (119)	3.02× (30)
SPECTRA R_K=640	6.7×	2.14× (412)	2.55× (106)	2.72× (26)
SPECTRA R_K=768	6.0×	1.92× (372)	2.24× (95)	2.49× (24)

Serving efficiency. Compression is only worth anything if the bytes it frees turn into work, so we measure throughput at a fixed memory budget rather than at a fixed batch size. On one H100 NVL we give the dense fp16 cache the largest batch the budget holds, we give SPECTRA the largest batch the same budget holds for its cache, and we time whole-model decode steps for both through the same attention kernel (Table 4). SPECTRA serves 1.9 to 3.4× the token per second of dense across the four ratios and the three context lengths we test, and the gain grows both with the compression ratio and with the context lengths. The reason for the second trend is that the cache is a larger share of what a decode step reads as the contexts grows, so the byte ratio governs a larger share of the step. The gain stay below the byte ratio because the weights and the feed-forward work do not shrink when the cache does, which bounds what any KV compression can buy at these lengths.

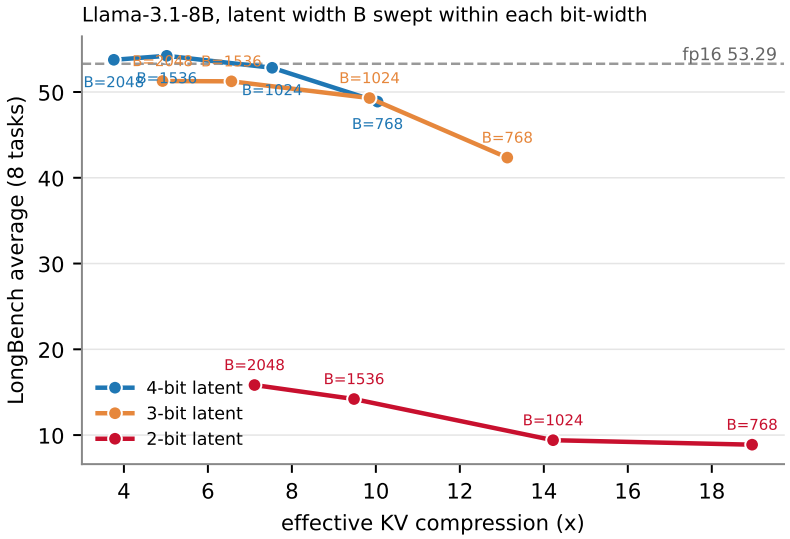

Figure 3: **Splitting the budget across bits and rank beats spending it all on bits** (Llama-3.1-8B, LongBench eight-task average, full test split). Effective compression against quality, sweeping the latent width B within each bit-width. At every matched compression the 3- and 4-bit curves dominate the 2-bit curve, so SPECTRA never operates on the cliff.

5 Conclusion

We have argued that the main lever in KV-cache compression is not how finely to quantize the keys and values, but what to store in the first place. The keys and values a model produces are highly redundant, yet that redundancy becomes usable only once they are rotated into a coordinate system computed from the model's own statistics, where a few channels carry almost all of it. SPECTRA turns this into a training-free, drop-in codec that gives each channel of that basis a bit-width matching its contribution, from several bits down to zero, so low-rank reduction and quantization become two settings of one control.

This pushes usable compression past the point where quantization alone collapses. Below about five times SPECTRA is level with the strongest quantizers. At eight times, where every uniform quantizer we compare against has fallen off the 2-bit cliff, SPECTRA still scores the dense average on Llama-3.1-8B, and it stays within a point of dense at eleven times. How far the cache compresses tracks how concentrated its energy is, most sharply on Qwen2.5-7B, where a cache half as wide costs about four extra points. Making that ceiling depend less on the model is the open problem this work leaves.

Reproducibility statement

Every number in this paper is measured on the full test split of the task it reports, and the setting behind each one is stated in the paper. The method is given in closed form in Section 3. Appendix B fixes the notation, Appendix C derives the weighted decomposition the method rests on, Appendix D derives the closed-form output-projection correction, and Appendix E derives the bit allocation. Appendix F states how the latent is read at attention time and what the cache holds. Appendix G names the harness, the tasks and the sample protocol, and it gives the accounting formula behind every compression ratio we report, including the metadata each ratio counts. Appendix H states the calibration corpus and the screen we used to accept or reject each optional trick. Appendix I gives the per-task LongBench scores and the full needle-in-a-haystack grids behind every averaged number in the main text.

AI USE STATEMENT

We used generative AI tools, in the form of an interactive coding assistant, to help implement our methods. This help covered checking for bugs, installing the environment, and verifying package versions. We also used these tools to check and revise the grammar of this paper. We did not use generative AI tools to develop the conceptual framework or the method itself, to formulate mathematical claims or to assist in proving them.

REFERENCES

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. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024a. URL https://arxiv.org/abs/2308.14508.

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, 2024b. URL https://arxiv.org/abs/2308.14508.

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.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2006.

DeepSeek-AI. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

Tim Dettmers and Luke Zettlemoyer. The case for 4-bit precision: k-bit inference scaling laws, 2023. URL https://arxiv.org/abs/2212.09720.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale, 2022. URL https://arxiv.org/abs/2208.07339.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. doi: 10.1007/BF02288367.

Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

V.K. Goyal. Theoretical foundations of transform coding. *IEEE Signal Processing Magazine*, 18(5): 9–21, 2001. doi: 10.1109/79.952802.

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.

R.M. Gray and D.L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44(6): 2325–2383, 1998. doi: 10.1109/18.720541.

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.

J. Huang and P. Schultheiss. Block quantization of correlated gaussian random variables. *IEEE Transactions on Communications Systems*, 11(3):289–296, 1963. doi: 10.1109/TCOM.1963.1088759.

Alan Julian Izenman. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248–264, 1975. ISSN 0047-259X. doi: https://doi.org/10.1016/0047-259X(75)90042-1. URL https://www.sciencedirect.com/science/article/pii/0047259X75900421.

Tao Ji et al. Towards economical inference: Enabling DeepSeek's multi-head latent attention in any transformer-based LLMs. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

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.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180.

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.

Shih-Yang Liu, Maksim Khadkevich, Nai Chit Fung, Charbel Sakr, Chao-Han Huck Yang, Chien-Yi Wang, Saurav Muralidharan, Hongxu Yin, Kwang-Ting Cheng, Jan Kautz, Yu-Chiang Frank Wang, Pavlo Molchanov, and Min-Hung Chen. Eora: Fine-tuning-free compensation for compressed llm with eigenspace low-rank approximation, 2026. URL https://arxiv.org/abs/2410.21271.

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. *arXiv preprint arXiv:2405.16406*, 2024a.

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.

Fanxu Meng et al. TransMLA: Multi-head latent attention is all you need. *arXiv preprint arXiv:2502.07864*, 2025.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

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.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023. URL https://arxiv.org/abs/2302.04761.

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.

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 Kępa, 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. *arXiv preprint arXiv:2403.07378*, 2024.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models, 2024a. URL https://arxiv.org/abs/2211.10438.

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

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, 2024. URL https://arxiv.org/abs/2405.15793.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

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.

Rongzhi Zhang et al. LoRC: Low-rank compression for LLMs KV cache with a progressive compression strategy. *arXiv preprint arXiv:2410.03111*, 2024.

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.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. Sglang: Efficient execution of structured language model programs, 2024. URL https://arxiv.org/abs/2312.07104.

Chenxi Zhou, Pengfei Cao, Jiang Li, Bohan Yu, Jinyu Ye, Jun Zhao, and Kang Liu. From signal degradation to computation collapse: Uncovering the two failure modes of llm quantization, 2026. URL https://arxiv.org/abs/2604.19884.

AI USE STATEMENT

We used generative AI tools, in the form of an interactive coding assistant, to help implement our methods. This help covered checking for bugs, installing the environment, and verifying package versions. We also used these tools to check and revise the grammar of this paper. We did not use generative AI tools to develop the conceptual framework or the method itself, to formulate mathematical claims or to assist in proving them.

A Extended Related Work

KV-cache quantization. The dominant approach to shrinking the cache lowers the bit-width of the stored keys and values. KIVI (Liu et al., 2024b) quantizes keys per-channel and values per-token to 2 bits, and KVQuant (Hooper et al., 2024) adds sensitivity-aware non-uniform quantization and outlier handling. A second line first *rotates* the cache to spread outliers before quantizing, using random or learned orthogonal transforms (QuaRot (Ashkboos et al., 2024), SpinQuant (Liu et al., 2024a), TurboQuant (Zandieh et al., 2025)) or a polar reparameterization (PolarQuant (Han et al., 2025)). All of these keep the cached object fixed to raw K, V and spend bits uniformly (or by heuristic importance) over its channels, and quality degrades sharply near 2 bits (the "2-bit cliff"). SPECTRA instead quantizes a second-moment-orthogonal, energy-ordered latent, where the measured marginal variances provide a canonical axis for rate allocation and low-rank truncation carries compression that scalar quantizers can only reach at INT2.

Token eviction and sparse attention. A complementary family reduces the *number* of cached tokens rather than the bits per token, keeping a budget of salient entries. H2O (Zhang et al., 2023) evicts by accumulated attention (heavy hitters), SnapKV (Li et al., 2024) selects tokens using a lookahead window, and StreamingLLM (Xiao et al., 2024b) retains attention sinks plus a sliding window. These methods are lossy on aggregation and multi-fact queries, where the discarded tokens are exactly the evidence a later query needs. SPECTRA is orthogonal: it compresses *every* token's representation rather than dropping tokens, and could be composed with eviction.

Low-rank and latent KV compression. Closest to us are methods that store a low-rank projection of the cache. Palu (Chang et al., 2024) caches a low-rank latent obtained from an SVD of the key/value projection weights, LoRC (Zhang et al., 2024) applies a progressive low-rank compression across layers, and SVD-LLM (Wang et al., 2024) and EoRA (Liu et al., 2026) use (truncation- or eigenspace-aware) low-rank factorization for LLM weight compression. The key difference is the *metric*: these factorize in weight space, ranking directions by weight energy, whereas SPECTRA minimizes error in the metric of the activations the model actually produces (G-weighted / KLT-like), which we show retains substantially more produced energy at equal rank and, unlike a pure whitening or random rotation, yields a latent that is both second-moment-orthogonal and energy-ordered.

Latent attention as an architecture. Multi-head latent attention (MLA) in DeepSeek-V2 (DeepSeek-AI, 2024) caches a compressed latent by design, and recent work retrofits existing models to MLA via architectural conversion and continued training/distillation (TransMLA (Meng et al., 2025), MHA2MLA (Ji et al., 2025)). SPECTRA targets the same latent-cache idea but as a *post-hoc, training-free* transform on an already-trained model: no pretraining, fine-tuning, or distillation, and a drop-in replacement of the key/value projections.

Transform coding and rate–distortion foundations. SPECTRA follows the classical transform-coding pattern (Goyal, 2001; Gray & Neuhoff, 1998): a data-dependent transform followed by bit allocation over its coefficients. The classical KLT diagonalizes the centered covariance of a source, whereas our G-weighted transform diagonalizes the uncentered latent second moment, after which we center the coefficients and allocate using their measured marginal variances. Reverse water-filling is rate–distortion optimal for parallel Gaussian sources (Huang & Schultheiss, 1963; Cover & Thomas, 2006; Gersho & Gray, 1992), the optimal low-rank factor in a weighted metric follows from Eckart–Young (Eckart & Young, 1936), and our output-side correction is a reduced-rank regression (Izenman, 1975). Our contribution is not these tools but their application to the KV cache: identifying G (the activation Gram matrix) as the right metric, and showing that transform coding in that metric turns the ill-posed problem of quantizing raw K, V into a well-posed one. The axis along which the budget is spent is what separates SPECTRA from the two families above. Eviction allocates over the token axis and cross-layer sharing allocates over the layer axis, whereas SPECTRA allocates over the channels of a transform fitted to the data.

B Notation

Bold lowercase denotes row vectors: a hidden state $\mathbf{h} \in \mathbb{R}^{1\times d}$, a key $\mathbf{k} \in \mathbb{R}^{1\times d_{kv}}$, a latent $\mathbf{c} \in \mathbb{R}^{1\times r}$. Bold uppercase denotes matrices: the projection $\mathbf{W}_K \in \mathbb{R}^{d\times d_{kv}}$, the stacked cache $\mathbf{K} = \mathbf{H}\mathbf{W}_K \in$

$\mathbb{R}^{L\times d_{kv}}$ over a length-L sequence, and the uncentered second-moment matrix $\mathbf{G} = \mathbb{E}_{\mathbf{h}}[\mathbf{h}^\top\mathbf{h}] \in \mathbb{R}^{d\times d}$. The operator $\mathbb{E}_{\mathbf{h}}$ is the expectation over hidden states on calibration data. We write B for the total latent width of a layer, r_k and r_v for the ranks it is split into, $\bar{b}$ for the average number of bits per latent coordinate, and g for the number of adjacent coordinates that share one bit-width.

C G-WEIGHTED SVD

Activation-metric identity. The same calculation underlies the objective of Section 3.1. For any matrix $\Delta\mathbf{W}$ (e.g. the factorization residual $\mathbf{W} - \mathbf{W}_{\text{down}}\mathbf{W}_{\text{up}}$), the expected error on produced activations is, with $\mathbf{h}$ a row vector,

$$\begin{aligned}\mathbb{E}_{\mathbf{h}}\|\mathbf{h}\,\Delta\mathbf{W}\|^2 &= \mathbb{E}_{\mathbf{h}}\,\mathrm{tr}\big(\Delta\mathbf{W}^\top\mathbf{h}^\top\mathbf{h}\,\Delta\mathbf{W}\big)\\ &= \mathrm{tr}\big(\Delta\mathbf{W}^\top\,\mathbb{E}_{\mathbf{h}}[\mathbf{h}^\top\mathbf{h}]\,\Delta\mathbf{W}\big)\\ &= \mathrm{tr}\big(\Delta\mathbf{W}^\top\mathbf{G}\,\Delta\mathbf{W}\big)\\ &= \big\|\mathbf{G}^{1/2}\Delta\mathbf{W}\big\|_F^2,\end{aligned} \tag{4}$$

using linearity of the trace and expectation and the symmetry $\mathbf{G}^{1/2\top}\mathbf{G}^{1/2} = \mathbf{G}$. Minimizing this over rank-r $\mathbf{W}_{\text{down}}\mathbf{W}_{\text{up}}$ is thus a plain low-rank problem on the whitened weight $\mathbf{G}^{1/2}\mathbf{W}$, solved in closed form by Equation 2.

Why the square-root split. Substituting $\mathbf{c} = \mathbf{h}\mathbf{W}_{\text{down}}$ into $\mathbb{E}[\mathbf{c}^\top\mathbf{c}]$ and using $\mathbb{E}_{\mathbf{h}}[\mathbf{h}^\top\mathbf{h}] = \mathbf{G}$ gives

$$\mathbb{E}[\mathbf{c}^\top\mathbf{c}] = \mathbf{W}_{\text{down}}^\top\,\mathbf{G}\,\mathbf{W}_{\text{down}} = \mathbf{\Sigma}_r \qquad \text{(diagonal, descending)}, \tag{5}$$

so the cached coordinates are orthogonal in uncentered second moment and ordered by energy. Every split of $\mathbf{\Sigma}_r$ between $\mathbf{W}_{\text{down}}$ and $\mathbf{W}_{\text{up}}$ reconstructs the same product, but only the square-root split produces this statistic. A pure whitening split ($\mathbf{W}_{\text{down}} = \mathbf{G}^{-1/2}\mathbf{U}_r$, $\mathbf{W}_{\text{up}} = \mathbf{\Sigma}_r\mathbf{V}_r^\top$) spans the same subspace and yields a unit-variance latent, which discards the energy ordering the bit allocator of Section 3.2 depends on. Equation equation 5 is not a covariance identity when $\mathbb{E}[\mathbf{c}] \neq 0$: before quantization we subtract the measured coordinate means and estimate the centered marginal variances separately.

G-weighted (ours). Let $\mathbf{M} = \mathbf{G}^{1/2}\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. Truncating $\mathbf{M}$ to rank r is optimal in produced energy (Eckart–Young (Eckart & Young, 1936) in the G-metric), so the retained fraction is $\sum_{j\le r}\sigma_j^2 / \sum_j \sigma_j^2$, where σ_j are the singular values of $\mathbf{M}$.

Plain SVD (Palu). Let $\mathbf{W} = \mathbf{U}_w\mathbf{\Sigma}_w\mathbf{V}_w^\top$ be the SVD of the weight itself, and keep its top-r left singular directions. Scored in the *same* produced-energy metric, the retained fraction is

$$\frac{\sum_{j\le r}\sigma_{w,j}^2\,\big(\mathbf{u}_{w,j}^\top\mathbf{G}\,\mathbf{u}_{w,j}\big)}{\|\mathbf{G}^{1/2}\mathbf{W}\|_F^2}, \tag{6}$$

which follows from $\|\mathbf{G}^{1/2}\mathbf{W}_r\|_F^2 = \sum_{j\le r}\sigma_{w,j}^2\,(\mathbf{u}_{w,j}^\top\mathbf{G}\,\mathbf{u}_{w,j})$ since the right singular vectors are orthonormal. Because this basis orders directions by weight energy $\sigma_{w,j}^2$ rather than by produced energy, it cannot retain more produced energy than the G-weighted optimum at equal rank and generally retains less.

Setup. $\mathbf{G}$ (and the mean $\bar{\mathbf{h}}$) are accumulated in fp64 via a forward hook on each layer's attention input, and the curves are averaged over all layers. The curves in this appendix are measured on 24 windows of length 2,048 from WikiText-2 train. That is a measurement setting for this figure only, and it is independent of the calibration SPECTRA uses at deployment, which Appendix G specifies.

D CLOSED-FORM $\mathbf{W}_O$ HEALING

The low-rank factorization of the value path perturbs the attention output, and the error reaches the residual stream through the output projection $\mathbf{W}_O$. Because $\mathbf{W}_O$ is not stored in the cache, we can compensate the dominant, structured part of this error by adding a small correction $\Delta\mathbf{W}$ to $\mathbf{W}_O$, at no inference cost. Crucially, the correction is fit on the *low-rank but un-quantized* model, so it heals the low-rank error, not the quantization noise (which is handled separately in Section 3.2).

Objective. Let $\mathbf{Y}$ be the input to $\mathbf{W}_O$ (the post-attention context) under the low-rank model and $\Delta\mathbf{O} = \mathbf{O}_{\text{orig}} - \mathbf{O}_{\text{lowrank}}$ the resulting output error, collected over calibration tokens. We want $\mathbf{Y}(\mathbf{W}_O + \Delta\mathbf{W}) \approx \mathbf{Y}\mathbf{W}_O + \Delta\mathbf{O}$, i.e. $\mathbf{Y}\,\Delta\mathbf{W} \approx \Delta\mathbf{O}$, with $\Delta\mathbf{W}$ constrained to rank ρ:

$$\min_{\operatorname{rank}\Delta\mathbf{W}\le\rho} \left\|\mathbf{Y}\,\Delta\mathbf{W} - \Delta\mathbf{O}\right\|_F^2. \tag{7}$$

Closed form. Let $\mathbf{A} = \mathbb{E}[\mathbf{Y}^\top\mathbf{Y}]$ and $\mathbf{B} = \mathbb{E}[\mathbf{Y}^\top\Delta\mathbf{O}]$ (both accumulated as streaming second moments, $\mathcal{O}(d^2)$ memory, no per-token buffers). The unconstrained least-squares solution is $\mathbf{A}^{-1}\mathbf{B}$, and the optimal rank-$\rho$ solution whitens, truncates, and un-whitens (Izenman, 1975):

$$\mathbf{M} = \mathbf{A}^{-1/2}\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \qquad \mathbf{M}_\rho = \mathbf{U}_{:\rho}\mathbf{\Sigma}_{:\rho}\mathbf{V}_{:\rho}^\top, \tag{8}$$

$$\Delta\mathbf{W}_\rho = \mathbf{A}^{-1/2}\mathbf{M}_\rho, \qquad \mathbf{W}_O \leftarrow \mathbf{W}_O + \Delta\mathbf{W}_\rho, \tag{9}$$

which is the optimal reduced-rank regression in the $\mathbf{A}$-metric. The SVD is computed once per layer, so sweeping ρ is essentially free, and in practice $\rho = 4$ recovers most of the lost quality at negligible parameter cost. The keys/values are not healed this way: the value error enters $\mathbf{W}_O$ linearly and is compensable there, whereas the key error acts through the softmax and is not.

E Reverse water-filling derivation

We derive the allocation of Equation 3, following classical optimal bit allocation (Huang & Schultheiss, 1963; Cover & Thomas, 2006). Under the scalar-source approximation of Section 3.2, treat the r centered latent coordinates as independent sources with measured marginal variances $v_1 \ge \cdots \ge v_r$. Under a high-rate scalar quantizer, coordinate j quantized at b_j bits incurs distortion $D_j = c\,v_j\,2^{-2b_j}$ for a constant c common to all coordinates. We minimize the total distortion at a fixed average rate:

$$\min_{b_1,\dots,b_r} \sum_{j=1}^{r} v_j\,2^{-2b_j} \quad \text{s.t.} \quad \frac{1}{r}\sum_{j=1}^{r} b_j = \bar{b}. \tag{10}$$

With a multiplier λ for the rate constraint, the Lagrangian is $\mathcal{L} = \sum_j v_j 2^{-2b_j} + \lambda\big(\sum_j b_j - r\bar{b}\big)$, and $\partial\mathcal{L}/\partial b_j = -2\ln 2\, v_j 2^{-2b_j} + \lambda = 0$ gives

$$v_j\,2^{-2b_j} \;=\; \frac{\lambda}{2\ln 2} \;=\; \theta \qquad \text{(a constant, the “water level”)}, \tag{11}$$

so the optimum equalizes the residual distortion D_j across all retained coordinates. Solving, $b_j = \frac{1}{2}\log_2(v_j/\theta)$. Imposing the rate constraint, $\frac{1}{r}\sum_j \frac{1}{2}\log_2(v_j/\theta) = \bar{b}$, i.e. $\log_2\theta = \frac{1}{r}\sum_j \log_2 v_j - 2\bar{b} = \log_2 \mathrm{GM}(v) - 2\bar{b}$ with $\mathrm{GM}(v) = (\prod_j v_j)^{1/r}$. Substituting back,

$$b_j = \bar{b} + \tfrac{1}{2}\log_2\big(v_j/\mathrm{GM}(v)\big), \tag{12}$$

which is Equation 3. The unconstrained b_j can be negative for low-variance coordinates, and clamping to $[0, b_{\max}]$ (reverse water-filling) sets those to zero bits, so the coordinate is dropped, which is the continuous bridge to rank reduction. The integer, group-level realization (per-group mean variance, integer rounding with a rate-matching fix-up) is described in Section 3.2.

F Latent-space attention and memory

Caching the latent opens a path that is closed to scalar quantizers, because the attention score is *linear* in the latent:

$$\mathbf{q}\,\mathbf{k}_t^\top \;=\; \mathbf{q}\,(\mathbf{c}_t\mathbf{W}_{\text{up}})^\top \;=\; (\mathbf{q}\,\mathbf{W}_{\text{up}}^\top)\,\mathbf{c}_t^\top \;=\; \mathbf{q}'\,\mathbf{c}_t^\top, \tag{13}$$

so the up-projection can be folded into the query once per decoding step ($\mathbf{q}' = \mathbf{q}\mathbf{W}_{\text{up}}^\top$, a cheap $1\times d_{kv} \to 1\times r$ map), and attention can then run *directly* in the r-dimensional latent space without ever materializing full keys or values. The value side is $\mathrm{out} = \mathrm{softmax}(\mathbf{q}'\mathbf{C}^\top)\,\mathbf{C}_V\,\mathbf{W}_{\text{up}}^V$, where only the final $\cdot\,\mathbf{W}_{\text{up}}^V$ lifts back to d_{kv}. The dominant matrix multiplies ($\mathbf{q}'\mathbf{C}^\top$ and $\mathrm{softmax}\cdot\mathbf{C}_V$) then act in dimension r rather than d_{kv}, so the memory saving becomes a compute saving as well. This is unavailable to KIVI/TurboQuant/PolarQuant, which cache $\mathbf{k}, \mathbf{v}$ directly and have no up-projection to absorb.

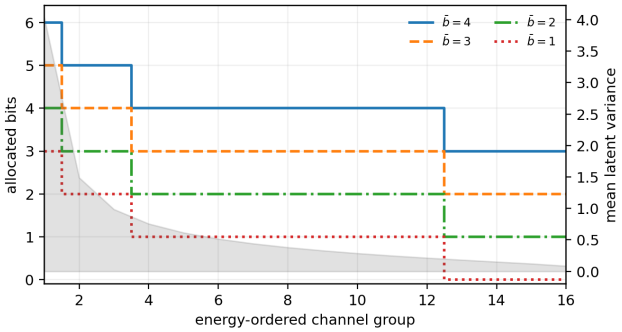

Figure 4: **Reverse water-filling on a real layer** (Llama-3.1-8B, layer 16, keys). Allocated bits (solid) track the latent energy spectrum (shaded): high-energy groups get more bits than the uniform baseline (dotted, KIVI's implicit choice), low-energy groups fewer. As $\bar{b}$=1, tail groups hit zero bits and quantization becomes rank reduction ($r_{\text{eff}} \approx 768/1024$).

G Evaluation and compression-accounting details

Calibration. We estimate the activation Gram matrices and means from 16 windows of 2,048 tokens, eight drawn from the WikiText-2 training split and eight from source code. Statistics are accumulated in fp64 from the input to each attention layer, and no evaluation example is used to construct the transform. Unless noted otherwise, the latent group size is $g = 64$ and the output correction has rank $\rho = 4$. SPECTRA-WF uses an equal rank split between keys and values ($\kappa = 0.5$), while SPECTRA-LR assigns one quarter of the total rank budget to keys ($\kappa = 0.25$). Every operating point is a setting of the two controls on bits and rank, the rank ceiling B and the average rate $\bar{b}$, and we identify each one in the tables by its measured compression rather than by an internal name. Qwen2.5-7B needs its own set of widths because its KV cache is half as wide as the other two models, so the range covered on Llama and Mistral is also covered there. The arXiv version of this work calibrated on WikiText-2 alone, and Appendix H reports both recipes at every operating point that was run under each.

Effective compression ratio. All reported compression ratios compare total KV-cache storage with the fp16 cache for the same model and sequence length. If b_{eff} denotes the effective number of stored bits per original KV element, including the payload, scales, and zero-points, then the reported ratio is $16/b_{\text{eff}}$. For SPECTRA-WF, $\bar{b}$ denotes the target average payload rate before metadata rather than a uniform channel bit-width. For SPECTRA-LR, the tables report the resulting effective ratio after both rank reduction and latent quantization. Baselines are accounted the same way, counting their per-group scales and zero-points, so all ratios are comparable. Two further points matter for reproducing our numbers. First, the +B configurations keep the first four and the last 64 tokens of $\mathbf{K}$ and $\mathbf{V}$ in fp16, and that window is charged. Because a window of fixed token count costs a share of the cache that depends on the sequence length, the reported ratio is byte-weighted over the examples actually evaluated, and it therefore shifts when the evaluation set changes. Second, a low-rank setting may request a latent wider than the KV cache, in which case the factor is capped at the KV width and we charge the capped width. Correcting this moved several ratios relative to the arXiv version, on Llama from $5.02\times$ to $5.51\times$ and from $6.56\times$ to $6.02\times$, and on Qwen it collapsed two settings onto a single $4.36\times$ operating point. Scores are unaffected.

Evaluation scope. LongBench uses the test split of `THUDM/LongBench` and the eight English tasks in Tables 7–9. **Every row is the full test split of every task.** We use the official prompt templates and per-task metrics, truncate inputs longer than 31,500 tokens by keeping equal-length prefix and suffix segments, apply the chat template to all tasks except the few-shot and code tasks (which the official protocol feeds without a wrapper), and decode greedily with the task-specific generation limits. Needle-in-a-Haystack plants unique magic codes on a grid of six context lengths (1k–32k) and seven depths, in single- and four-needle variants, and scores exact recovery under greedy decoding. Every method, dense, SPECTRA, and all baselines alike, is evaluated through this single harness.

H CALIBRATION CORPUS AND THE TRICK SCREEN

SPECTRA's transform is fitted to activation statistics, so the corpus those statistics come from is a real design choice. The arXiv version of this work drew all calibration windows from WikiText-2. This submission draws half of them from source code, and Table 5 reports both recipes at every operating point of every model, with everything else held fixed. The two agree within a point almost everywhere, and sometimes the WikiText-only recipe is slightly ahead. The exception is the most aggressive operating point, where a code-bearing corpus matters a great deal: on Llama at $11.09\times$ the average rises from 47.90 to 52.44, driven by LCC and RepoBench-P, and on Mistral the same point gains 1.22. The reading we take from this is that at mild compression the transform has enough rank to serve any input distribution, while at aggressive compression the retained directions are few enough that the calibration corpus decides which inputs survive.

Table 5: **Calibration corpus ablation.** LongBench average (eight tasks, full test split) with the arXiv calibration (WikiText-2 only) and the conference recipe (WikiText-2 + code, 8+8 windows of 2,048 tokens), at every operating point that was run under both. Everything else is identical. On Qwen2.5-7B two operating points request the same effective latent width once the rank cap of Appendix G is applied, so they are listed once.

Model	Comp.	WikiText	+code	Δ
Llama-3.1-8B	$3.56\times$	53.08	53.18	$+0.10$
	$3.95\times$	53.12	52.94	-0.17
	$4.57\times$	52.72	52.62	-0.09
	$5.51\times$	52.33	51.74	-0.60
	$5.75\times$	52.28	51.95	-0.34
	$6.02\times$	52.47	52.59	$+0.13$
	$6.74\times$	52.48	52.88	$+0.41$
	$7.65\times$	53.80	53.29	-0.51
	$8.82\times$	52.75	52.99	$+0.24$
	$11.09\times$	47.90	52.44	$+4.54$
Mistral-7B	$3.56\times$	53.08	52.81	-0.27
	$3.97\times$	52.13	52.94	$+0.81$
	$4.57\times$	52.41	52.95	$+0.53$
	$5.55\times$	52.82	51.93	-0.89
	$5.80\times$	52.34	52.05	-0.29
	$6.07\times$	52.32	51.76	-0.57
	$6.81\times$	51.92	52.30	$+0.39$
	$7.73\times$	51.34	52.07	$+0.73$
	$8.93\times$	51.16	52.32	$+1.16$
	$11.27\times$	49.54	50.76	$+1.22$
Qwen2.5-7B	$3.48\times$	53.51	52.91	-0.60
	$4.36\times$	52.67	53.54	$+0.87$
	$4.43\times$	52.63	52.62	-0.01
	$4.91\times$	51.09	51.73	$+0.64$
	$4.92\times$	51.58	52.27	$+0.70$
	$5.64\times$	51.09	50.84	-0.25

The calibration corpus was chosen by a screen, not by search over the test set. Table 6 varies one ingredient at a time against the arXiv recipe on Llama, at a moderate and an aggressive operating point, scored on four LongBench tasks at full test split. Of the four candidates, only the calibration corpus helps at both points, and it is the only one this submission adopts. Clipping the latent and enlarging the output correction beyond the shipped rank $\rho{=}4$ both cost quality at one point or the other, which is what overfitting a small calibration set looks like. A rank-32 key-side residual patch is close to free in quality and costs compression, so it is not worth its bytes here.

Table 6: **Only the calibration corpus was worth adopting.** One variable at a time on Llama-3.1-8B at two operating points, scored on four LongBench tasks at full test split (Qasper, TriviaQA, LCC, RepoBench-P). Δ is against the control, which is the recipe of the arXiv version. Adding code to the calibration corpus is the only change that helps at both points, and it is the only one this submission adopts. Enlarging the output correction beyond the shipped rank ρ=4 costs quality, which is consistent with it overfitting the calibration set.

Point	Variant	Comp.	Mean	Δ
moderate	control (paper recipe)	8.82×	63.77	–
	calibration + code	8.79×	64.08	+0.32
	clip 0.95	8.79×	63.93	+0.17
	clip 0.98	8.79×	62.75	−1.01
	K-whisper r=16	8.24×	63.53	−0.24
	K-whisper r=32	7.74×	63.63	−0.13
	$\mathbf{W}_O$ heal rank 16	8.79×	62.94	−0.83
	$\mathbf{W}_O$ heal rank 32	8.79×	60.95	−2.81
aggressive	control (paper recipe)	11.09×	54.84	–
	calibration + code	11.05×	63.79	+8.95
	clip 0.95	11.05×	53.92	−0.92
	clip 0.98	11.05×	54.37	−0.47
	K-whisper r=16	10.18×	54.91	+0.07
	K-whisper r=32	9.44×	55.35	+0.52
	$\mathbf{W}_O$ heal rank 16	11.05×	56.07	+1.23
	$\mathbf{W}_O$ heal rank 32	11.05×	54.75	−0.09

I Full Experiment Results

I.1 LongBench: Per-Task Results

Every SPECTRA operating point and every baseline configuration, per task, on the full test split. The abbreviated Llama table of Section 4.2 is one row per tier drawn from Table 7.

Table 7: **LongBench per task, Llama-3.1-8B-Instruct.** Full sweep: every SPECTRA operating point and all baselines, grouped by compression regime. Avg is the mean over the eight tasks and every row is the full test split. Within each tier the **best** and second Avg are highlighted.

Method	Comp.	Qasper	QMSum	MNews	TREC	TQA	SAMSum	LCC	RB-P	Avg
base (fp16)	1.00×	46.57	25.19	27.03	72.50	91.65	43.47	63.38	56.55	**53.29**
SPECTRA	3.56×	46.14	25.57	26.90	72.50	92.17	43.65	63.42	55.08	53.18
SPECTRA	3.95×	47.46	25.41	26.27	72.00	90.81	43.16	63.65	54.79	52.94
PolarQuant r4t4v4	3.66×	46.01	25.56	26.69	72.50	91.86	42.88	65.38	58.44	53.66
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
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
SPECTRA	4.57×	44.48	25.04	27.02	71.50	90.90	43.43	63.14	55.47	**52.62**
SPECTRA	5.51×	43.40	25.32	26.61	72.00	89.66	43.67	60.57	52.66	51.74
Palu r0.9 4-bit	4.41×	45.12	24.67	26.63	73.00	92.15	44.73	59.72	52.11	52.27
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
SPECTRA	5.75×	43.62	24.83	26.61	70.00	90.93	44.12	60.78	54.68	51.95
SPECTRA	6.02×	43.85	25.07	26.26	71.50	91.15	44.15	62.07	56.70	52.59
SPECTRA	6.74×	44.20	25.51	25.83	70.50	90.14	44.11	61.83	60.95	52.88
PolarQuant r2t2v2	6.74×	41.66	24.03	23.54	59.00	92.25	38.88	58.23	50.46	48.51
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**
SPECTRA	7.65×	45.34	25.12	26.17	71.50	91.78	43.40	61.77	61.26	**53.29**
Palu r0.7 3-bit	7.51×	36.22	24.07	24.86	69.50	86.60	42.49	49.79	50.49	48.00
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
SPECTRA	8.82×	45.84	25.24	25.69	73.00	89.77	43.64	62.77	57.95	**52.99**
SPECTRA	11.09×	47.00	23.75	25.78	69.50	88.60	44.23	63.88	56.81	52.44

Table 8: **LongBench per task, Mistral-7B-Instruct-v0.3.** Full sweep: every SPECTRA operating point and all baselines, grouped by compression regime. Avg is the mean over the eight tasks and every row is the full test split. Within each tier the **best** and second Avg are highlighted.

Method	Comp.	Qasper	QMSum	MNews	TREC	TQA	SAMSum	LCC	RB-P	Avg
base (fp16)	1.00×	38.15	25.57	26.43	76.00	88.59	47.46	59.46	60.56	**52.78**
SPECTRA	3.56×	38.47	25.74	26.46	76.00	88.02	47.46	59.09	61.24	52.81
SPECTRA	3.97×	38.59	25.70	26.54	76.00	88.87	47.57	59.32	60.94	52.94
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
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
SPECTRA	4.57×	37.38	25.65	26.75	75.50	89.05	47.05	60.37	61.84	**52.95**
SPECTRA	5.55×	38.40	25.46	26.24	75.50	87.42	46.04	57.00	59.35	51.93
Palu r0.9 4-bit	4.40×	38.67	25.35	26.70	75.50	88.10	47.08	60.43	60.30	52.77
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
SPECTRA	5.80×	38.54	25.42	26.21	76.50	88.29	46.01	57.26	58.16	52.05
SPECTRA	6.07×	38.09	24.60	26.15	77.00	87.80	45.88	55.90	58.65	51.76
SPECTRA	6.81×	36.94	25.33	26.36	75.50	89.23	47.01	58.27	59.78	52.30
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**
SPECTRA	7.73×	36.59	25.63	26.56	76.00	89.47	47.07	57.37	57.87	**52.07**
Palu r0.7 3-bit	7.50×	35.22	24.08	25.83	71.00	87.34	44.61	56.36	57.07	50.19
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
SPECTRA	8.93×	38.67	25.72	26.61	75.00	89.10	46.80	58.86	57.78	**52.32**
SPECTRA	11.27×	36.75	25.85	26.22	71.50	87.55	45.60	56.31	56.26	50.76

Table 9: **LongBench per task, Qwen2.5-7B-Instruct.** Full sweep: every SPECTRA operating point and all baselines, grouped by compression regime. Avg is the mean over the eight tasks and every row is the full test split. Within each tier the **best** and second Avg are highlighted.

Method	Comp.	Qasper	QMSum	MNews	TREC	TQA	SAMSum	LCC	RB-P	Avg
base (fp16)	1.00×	43.52	23.61	23.98	71.50	89.64	45.98	60.77	66.82	**53.23**
SPECTRA	3.48×	43.26	23.81	24.11	71.50	88.85	45.52	59.35	66.90	52.91
PolarQuant r4t4v4	3.66×	43.94	24.42	25.52	72.00	89.46	45.76	63.68	66.41	**53.90**
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
SPECTRA	4.36×	45.15	23.55	24.30	71.50	88.93	46.65	62.59	65.63	**53.54**
SPECTRA	4.43×	44.66	23.36	23.61	71.50	88.81	44.91	59.92	64.16	52.62
SPECTRA	4.91×	42.32	23.38	24.01	68.00	89.49	45.14	60.69	60.78	51.73
SPECTRA	4.92×	42.96	23.40	23.05	70.00	88.83	46.04	59.76	64.14	52.27
Palu r0.9 4-bit	4.37×	42.55	23.87	24.16	72.50	86.86	45.10	57.65	63.81	52.06
PolarQuant r2t2v4	4.74×	38.51	22.70	24.87	69.00	89.54	44.25	60.61	57.45	50.87
TurboQuant K4V2	5.33×	42.56	23.27	23.53	72.00	88.29	45.30	55.14	59.88	51.25
SPECTRA	5.64×	39.11	23.66	24.09	71.50	86.77	46.26	60.47	54.90	**50.84**
PolarQuant r2t2v2	6.74×	37.63	23.09	25.41	67.50	88.08	44.44	58.11	56.74	50.12
SPECTRA	7.65×	31.01	23.42	25.52	70.00	89.69	45.17	53.04	51.87	**48.72**
SPECTRA	8.00×	32.53	23.16	25.14	72.50	89.48	44.04	50.15	50.74	48.47
SPECTRA	8.00×	31.05	23.04	25.58	70.50	90.11	43.49	53.40	51.49	48.58
Palu r0.7 3-bit	7.37×	32.84	23.95	25.27	73.00	86.61	44.76	51.31	50.24	48.50
TurboQuant K2V2	8.00×	36.18	21.85	22.25	66.50	81.39	42.92	55.44	50.93	47.18
SPECTRA	8.82×	34.18	21.85	25.24	70.00	88.20	43.29	52.13	46.50	**47.67**

I.2 NEEDLE-IN-A-HAYSTACK (NIAH)

Table 10: **Needle-in-a-haystack, overall retrieval (%).** Six context lengths (1k–32k) × seven depths, scoring a single needle by whether it is recovered and four needles by the fraction recovered. Every method runs through the same harness.

Model	Method	Single	Four
Llama-3.1-8B	dense (fp16)	100.0	85.71
	SPECTRA (6.6×)	100.0	85.12
	SPECTRA (12.2×)	100.0	100.0
	SPECTRA (3.6×)	100.0	84.52
	SPECTRA (9.5×)	100.0	94.64
	KIVI K2V2 (7.1×)	100.0	93.45
	OTT K2V2 (7.1×)	100.0	88.1
	PolarQuant r2t2v4 (mixed) (4.7×)	50.0	42.86
	RotateKV 2-bit (8.0×)	97.62	81.55
	TurboQuant K2V2 (8.0×)	88.1	63.1
Mistral-7B	dense (fp16)	100.0	66.67
	SPECTRA (6.6×)	100.0	65.48
	SPECTRA (12.2×)	100.0	70.83
	SPECTRA (3.6×)	100.0	66.67
	SPECTRA (9.5×)	100.0	67.86
	KIVI K2V2 (7.1×)	92.86	52.38
	OTT K2V2 (7.1×)	95.24	56.55
	RotateKV 2-bit (8.0×)	100.0	56.55
	TurboQuant K2V2 (8.0×)	66.67	47.02
Qwen2.5-7B	dense (fp16)	100.0	76.19
	SPECTRA (6.1×)	100.0	76.19
	SPECTRA (3.6×)	100.0	75.6
	SPECTRA (4.6×)	100.0	75.6
	SPECTRA (4.9×)	100.0	78.57
	PolarQuant r2t2v4 (mixed, bias-free keys) (4.7×)	95.24	74.4
	TurboQuant K2V2 (8.0×)	100.0	66.67

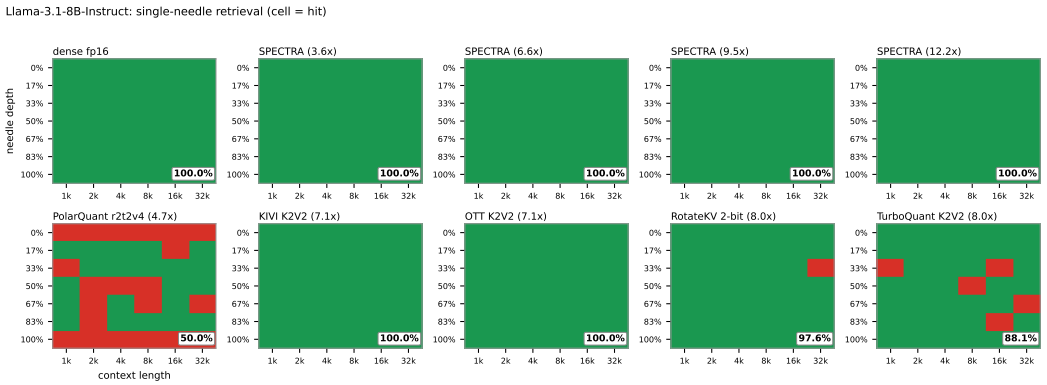

Figure 5: **Single-needle NIAH, Llama-3.1-8B-Instruct.** Retrieval accuracy across context length (horizontal) and needle depth (vertical), where a filled cell is a recovered needle and the badge reports overall accuracy. SPECTRA matches the dense model at every tier tested, while the 2-bit baselines start losing cells at the same compression or less.

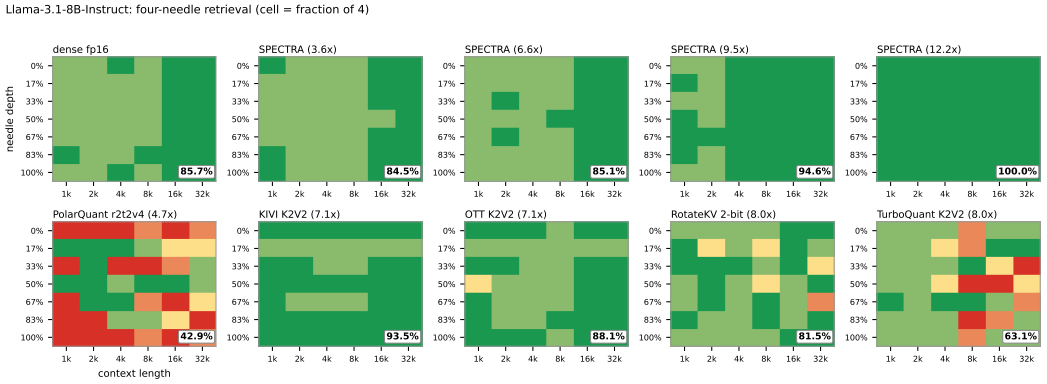

Figure 6: **Four-needle NIAH, Llama-3.1-8B-Instruct.** Fraction of the four planted needles recovered over the same context-by-depth grid, with the mean fraction in the badge. The harder variant separates the methods more than the single-needle one does.

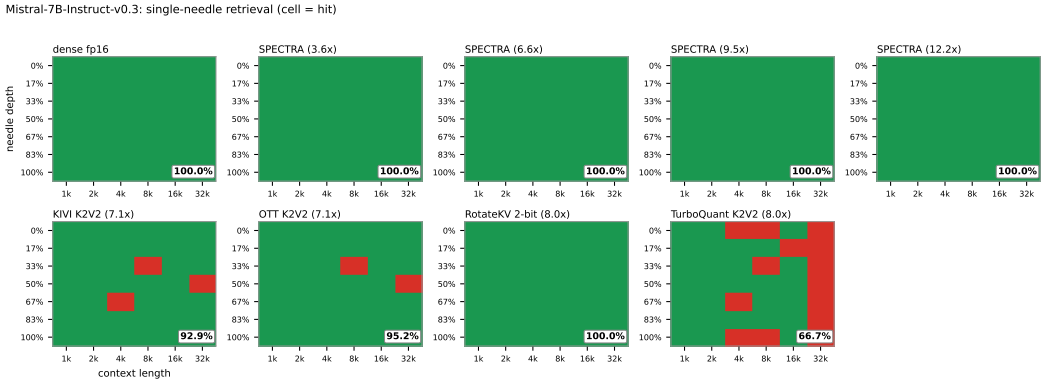

Figure 7: **Single-needle NIAH, Mistral-7B-Instruct-v0.3.** Axes and scoring as in Figure 5.

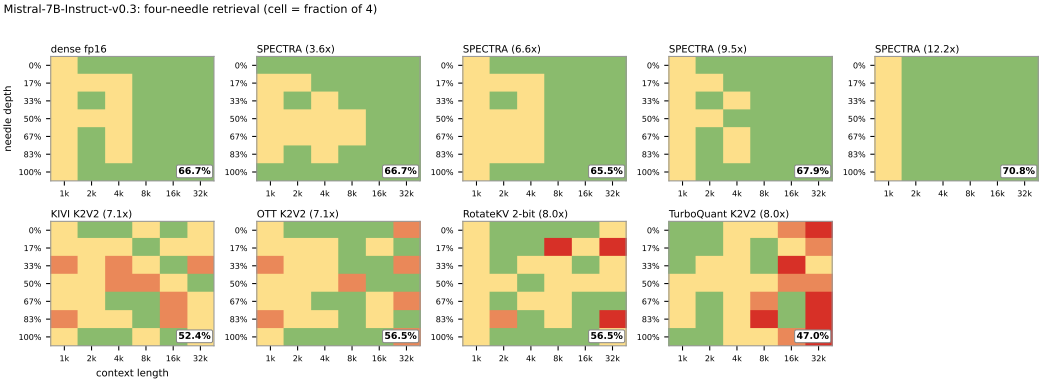

Figure 8: **Four-needle NIAH, Mistral-7B-Instruct-v0.3.** Axes and scoring as in Figure 6.

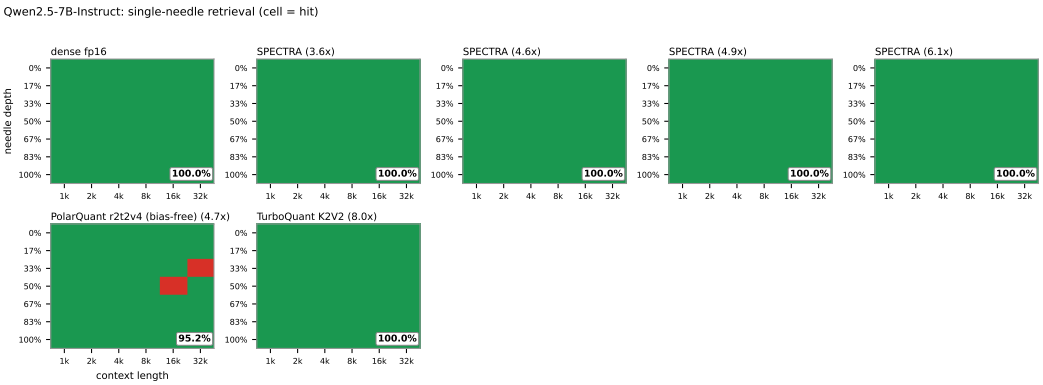

Figure 9: **Single-needle NIAH, Qwen2.5-7B-Instruct.** Axes and scoring as in Figure 5.

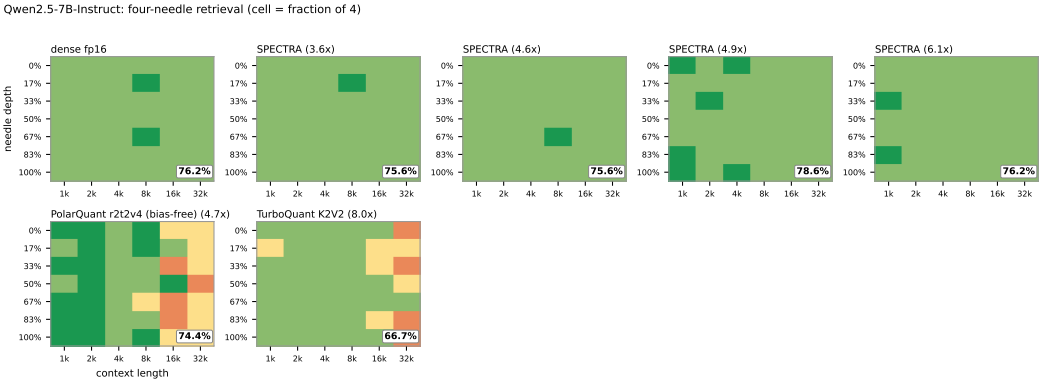

Figure 10: **Four-needle NIAH, Qwen2.5-7B-Instruct.** Axes and scoring as in Figure 6. SPECTRA stays within a point of the dense model at every tier tested, sometimes just above it and sometimes just below.