OPTQUANT: QUANTIZATION ERROR IS RELOCATED, NOT ELIMINATED

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training quantization cannot remove the error it introduces; it can only decide where that error goes. Rounding to a fixed grid injects a residual whose size is set by the grid alone, identical for every algorithm, so what a quantizer actually controls is not the magnitude of its error but its placement across channels, basis directions and time. We formalize this view. For any linear operator that consumes a quantized tensor, the distortion that survives into the output is governed by that operator's spectrum, and a good quantizer is one that steers error into the direc- tions the operator attenuates. The framework prescribes an operator per axis and, equally important, predicts which axes are not worth treating: it rejects the tem- poral axis of the value cache from a measured lag-1 attention autocorrelation of about 0.05 before any experiment is run. Instantiating it yields a calibration-free sequential quantizer for activations, whose downstream operator is given exactly by the layer's own weights rather than estimated from data, and a composite of channel-mean removal and head-wise Hadamard rotation for the key cache. On W4A4KV4 the training-free configuration alone reaches 5.85 perplexity on Llama- 2-7B, ahead of QuaRot (6.12), FlatQuant (6.08), SpinQuant (5.98) and OSTQuant (5.95), all of which learn their transforms; adding a transform learned through the relocation operator reaches 5.714 ±0.007 and 6.899 on Llama-3-8B, ahead of the previous state of the art. Measured in the downstream eigenbasis, rotation shrinks total error tenfold but leaves its placement at the white-noise level, whereas relo- cation moves 12.4% of error out of the most damaging directions down to 0.7%, which is why the rotation family plateaus where it does.

1 INTRODUCTION

The serving cost of large language models (LLMs) is no longer set by weight memory alone: as contexts lengthen and batches grow, activations and key-value (KV) caches bottleneck both memory capacity and bandwidth. Post-training quantization (PTQ) research has accordingly converged on `W4A4KV4` as a de facto target, a regime where naive round-to-nearest (RTN) collapses and where orthogonal rotations, per-channel scaling and learnable equivalence transformations have been proposed as remedies.

Disparate as they look, these techniques rest on one arithmetic fact: *once the grid and scale are fixed, quantization error is not eliminated but relocated.* With a fixed budget of discrete states, no algorithm reduces the noise injected per element by rounding onto a fixed lattice. GPTQ nonetheless beats RTN, not by shrinking aggregate error but by moving it. The same error magnitude is catastrophic when aligned with directions a downstream operator amplifies and harmless when steered into directions it attenuates. Seen this way every prevailing technique performs *implicit error relocation*: GPTQ pushes rounding residuals onto unquantized columns, SmoothQuant shifts difficulty from activations to weights, rotations disperse outlier energy across channels. Which axis to steer along, however, has remained an empirical choice; no framework derives the target directions a priori from the tensor and its consuming operator, or forecasts the gain.

We *formalize* that choice: where to put error is not a heuristic to tune but an objective to optimize. Once the downstream linear operator consuming the quantized tensor is named, the distortion surviving it is a deterministic quantity fixed by that operator's spectrum, and quantizer design becomes the minimization of a formal objective over *where* error resides. We call the principle *matched error*

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.

relocation and the quantizer it prescribes **OptQuant** (optimized quantization). Minimizing element-wise error is optimal only when the operator is isotropic; transformer primitives are not, and their spectra are often computable analytically, with no calibration data.

The formalization pays twice. First it *derives* the operator for each axis: on the channel axis the weight matrix the activations traverse defines downstream distortion exactly, yielding a calibration-free sequential activation quantizer; on the rotation axis, key-cache outliers turn out to be dominated by stationary channel offsets, which motivates channel-mean centering composed with per-head Hadamard rotations. Second, it names the *axes not worth pursuing*: the value cache's temporal axis is ruled out before any trial from a measured lag-1 autocorrelation of $\bar{\rho} \approx 0.05$, and the prediction that value-cache rotation binds only at 2 bits is then confirmed. In the downstream eigenbasis the contrast is stark: rotations are memoryless, suppressing error magnitude while leaving its placement nearly white, whereas relocation controls the *subspace* error projects onto, which explains the plateau of rotation-based methods. Since every candidate axis's gain is computable from operator statistics before a tensor is rounded, futile axes are rejected up front; and since the objective is quadratic, each axis has a closed form with no learnable parameters, which is why our calibration-free, tuning-free configuration already outperforms learned transformations.

Summary of Contributions. (i) **Formalization of error relocation.** We unify weight, activation and KV quantization into one distortion objective under downstream operators, proving that the gain available on each axis is governed by the operator's spectral anisotropy, so viable axes are targeted and futile ones pruned before any experiment (e.g. the temporal axis, $\bar{\rho} \approx 0.05$). (ii) **Calibration-free sequential activation quantization.** Because activation distortion is determined analytically by the weights, sequential rounding needs no calibration data; with layer-wise compensation matrices and block approximations it captures $\sim 95\%$ of the theoretical bound at a $\sim 100\times$ speedup. (iii) **Decoupled KV-cache distortion.** Channel-mean centering and per-head Hadamard rotations remove mutually orthogonal error components in key caches, giving near-lossless 3-bit keys (5.92 vs. 5.73 FP16 PPL), and trace the 2-bit value-cache bottleneck to heavy-tailed per-head crest factors (median 2.95, 99th percentile 10.19). (iv) **Relocation-aware transformation learning (ROTA-STE).** We optimize equivalence transformations *through* the forward dynamics of the relocation operator rather than around memoryless rounding. At `W4A4KV4` the tuning-free analytic configuration already reaches 5.85 PPL on Llama-2-7B against QuaRot (6.12), SpinQuant (5.98) and OSTQuant (5.95); ROTA-STE reaches 5.714 ± 0.007 on Llama-2-7B and 6.899 on Llama-3-8B, past the best reported figures (ReSpinQuant 5.74/7.24), while retaining 91% of FP16 MMLU. These are the strongest `W4A4KV4` results we are aware of, and they need no network-wide rotation training. These comparisons are against cited baseline figures whose cache granularity, harness and hardware were outside our control (Appendix C.14).

2 MOTIVATION

Reading a quantizer not as a discretization operator but as an *explicit decision about which geometric subspace to relocate error into* explains, in one frame, both why existing post-training quantization (PTQ) methods work and why they have recently plateaued. We give four empirical observations and then formalize the design problem.

Observation 1. Rounding error magnitude is fixed, and the two available levers are mutually orthogonal. A b-bit uniform quantizer injects noise of element-wise variance $\Delta^2/12$; once the grid spacing is set that magnitude is an arithmetic invariant, not an algorithmic choice. A quantizer therefore has exactly two degrees of freedom: **magnitude control**, which shrinks Δ by transforming the basis, and **directional relocation**, which herds the fixed residual energy into the attenuating subspace of the downstream operator. They act on different quantities and so are decoupled. Rotating before rounding cuts *downstream distortion* $\|\mathbf{W}(\mathbf{x} - \hat{\mathbf{x}})\|^2$ by an order of magnitude yet leaves the *share* of error energy in the most sensitive top-10% eigenspace at its isotropic baseline: it lowers the carpet without changing its shape. Sequential relocation runs the other way, halving downstream distortion while *inflating* raw error energy (§4.6). Neither substitutes for the other, and rotation-based frameworks (Ashkboos et al., 2024b; Liu et al., 2025; Hu et al., 2025; Sun et al., 2025) have plateaued because they use only the first.

Observation 2. Downstream operators are anisotropic, and known in closed form along the activation path. A quantized tensor is consumed by a downstream linear operator $\mathbf{A}$, so the distortion *that operator passes forward* is exactly the quadratic form $\mathcal{D} = \mathbb{E}[(\mathbf{x} - \hat{\mathbf{x}})^\top \mathbf{H}(\mathbf{x} - \hat{\mathbf{x}})]$ with $\mathbf{H} = \mathbf{A}^\top \mathbf{A}$, whose eigenspectrum names both the directions that absorb error and those where it must not land. This concerns the operator's immediate output, not the end-to-end task loss; residual connections, layer normalizations and attention non-linearities sit between the two, so we treat the coupling to task-level perplexity as an empirical premise and verify it in §4.6. Element-wise MSE is optimal iff $\mathbf{H} \propto \mathbf{I}$, but across transformer primitives $\mathbf{H}$ departs from flat by over three orders of magnitude (Appendix Figure 5). On the activation path $\mathbf{H} = \mathbf{W}^\top \mathbf{W}$ is fixed by the static weights, so, unlike GPTQ's $\mathbb{E}[\mathbf{X}\mathbf{X}^\top]$, which must be estimated from samples, our activation compensation is entirely data-free (§3.2).

Observation 3. Existing methods already relocate error, but each follows an ad-hoc choice of a single axis. GPTQ propagates rounding residuals onto unquantized channels (channel axis), SmoothQuant (Xiao et al., 2023) and AWQ (Lin et al., 2024) move dynamic-range difficulty from activations to weights by diagonal equivalence transformations (scale axis), and rotation-based methods disperse spatial outliers uniformly (basis axis). None states quantization as a unified relocation objective, so none can say in advance which subspaces of a downstream operator attenuate error, or what a candidate axis is worth before quantizing, leaving axis selection to trial and error that spends compute on ineffective dimensions and forfeits high-gain ones.

Observation 4. Achievable gains along each axis are predictable a priori, and vary across tensors. Within one attention block, keys (K) and values (V) demand opposite treatments. K's outliers are dominated by stationary per-channel DC offsets (median per-head crest factor 6.31); subtracting a calibration-derived mean vector cuts this to 3.27, and composing that centering with a per-head Hadamard makes 3-bit keys virtually lossless (5.92 vs. 5.73 FP16 PPL). V behaves inversely: the median head is near-Gaussian (crest factor 2.95) while the top 1% forms a heavy tail with a 99th percentile of 10.19, so any head-averaged statistic calls V well-behaved and per-head processing is required. The temporal axis is rejected outright: the ceiling for Σ-Δ modulation is set by the lag-1 autocorrelation of attention weights, negligible in autoregressive decoders ($\bar{\rho} \approx 0.05$), so the operator is better omitted than tuned. OptQuant thus derives a different operator per path (**exact closed-form compensation**, **mean centering composed with rotation**, and **deliberate pruning of futile axes**) from statistics measured a priori rather than from post-hoc tuning. The design problem is then: given a tensor and its consuming operator, the eigenspectrum of $\mathbf{H} = \mathbf{A}^\top \mathbf{A}$ says where error can go, each degree of freedom's gain is read off that spectrum *before* quantizing, and an operator is deployed only where its gain is provably non-trivial. §3 solves this in closed form on three axes and §4 tests the predictions; Appendix C.5 extends these four observations.

3 METHOD

Let $\mathcal{Q}(\cdot)$ be a b-bit uniform round-to-nearest quantizer, $\hat{\mathbf{x}}$ the quantization of $\mathbf{x}$, and $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$ the residual. Following §2, each tensor is optimized under the **distortion metric** induced by its consuming downstream linear operator $\mathbf{A}$:

$$\mathcal{D}(\hat{\mathbf{x}}) = \mathbb{E}\left[\|\mathbf{A}(\mathbf{x} - \hat{\mathbf{x}})\|_2^2\right] = \mathbb{E}\left[(\mathbf{x} - \hat{\mathbf{x}})^\top \mathbf{H}(\mathbf{x} - \hat{\mathbf{x}})\right], \quad \mathbf{H} = \mathbf{A}^\top \mathbf{A} + \delta \cdot \mathrm{mean}(\mathrm{diag}(\mathbf{A}^\top \mathbf{A})) \cdot \mathbf{I}, \tag{1}$$

where δ is a ridge damping coefficient (λ is reserved for eigenvalues and for the temporal feedback gain of §3.5). We build a generic relocation operator minimizing equation 1, specialize it to the channel, rotation and temporal axes, and then formulate transformation learning through that operator. Figure 1 shows where each operator sits; Appendix C.10 gives the full pipeline.

3.1 METRIC-MATCHED RELOCATION OPERATOR

A greedy solution to equation 1 is obtained by sequentially quantizing coordinates while propagating accumulated residuals onto unquantized coordinates via the upper-triangular Cholesky factor $\mathbf{U}$ of $\mathbf{H}^{-1}$:

$$e_j = \frac{x_j - \mathcal{Q}(x_j)}{U_{jj}}, \quad x_k \leftarrow x_k - e_j \cdot U_{jk} \quad (k > j). \tag{2}$$

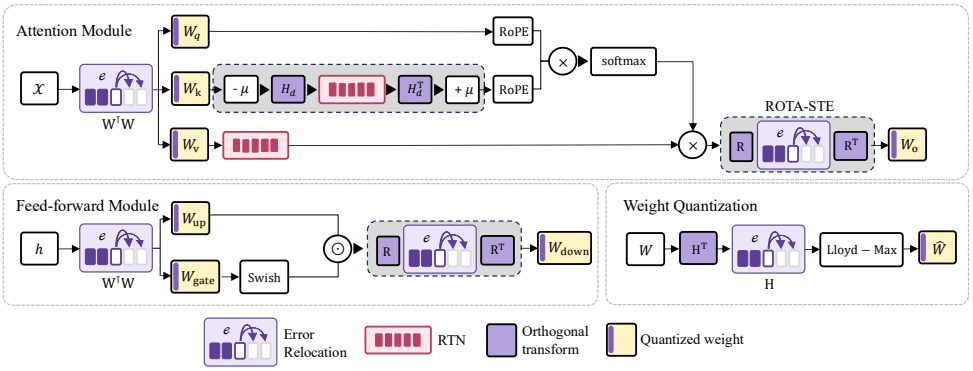


Figure 1: System architecture detailing operator placement and downstream consumption. Every operator is explicitly matched to the downstream linear operator it feeds. Orthogonal rotations are applied and inverted strictly within each linear layer, preserving the original basis of the residual stream.

This update is the algorithmic engine of GPTQ (Frantar et al., 2023); we generalize it past weights into a **universal operator** $\mathcal{Q}^{\mathbf{H}}_{\text{reloc}}(\cdot)$ acting on any tensor under any downstream metric $\mathbf{H}$. Its efficacy is set by the conditioning of $\mathbf{H}$, which is what the preprocessing below addresses.

Sequential relocation replaces the diagonal of $\mathbf{H}$ with the pivots of its $UDU^\top$ factorization (equivalently, $LDL^\top$ pivots in reversed coordinate order, matching the upper-triangular factor used above). Each pivot is a Schur complement and so never exceeds the diagonal entry it replaces, hence expected distortion never increases. The achievable gain is bounded by the arithmetic-to-geometric mean ratio of the eigenvalues of $\mathbf{H}$, a ceiling *invariant* under orthogonal transformation of the input dimension (Proposition 1, Appendix A.1). Transformations cannot raise that headroom; they equalize the pivots and push the realized gain toward it. This is why preprocessing targets $\mathbf{H}$ directly, and why the choice among incoherent transforms barely matters (§4.5: a structured Hadamard buys -0.017 PPL over a random orthogonal matrix, about 5% of the gain).

3.2 Channel Axis: Weight-Matched Activation Quantization

For activations entering $\mathbf{y} = \mathbf{W}\mathbf{x}$ the downstream operator is $\mathbf{W}$ itself, so equation 1 is available in closed form as $\mathbf{H} = \mathbf{W}^\top\mathbf{W}$. Being exact, time-invariant and shared by all tokens, it needs no calibration data: $\mathbf{U}$ is computed once per layer offline and the traversal in equation 2 vectorizes along the token dimension (Proposition 2, Appendix A.2), unlike GPTQ weight quantization, where $\mathbf{H} = \mathbb{E}[\mathbf{x}\mathbf{x}^\top]$ must be estimated from samples.

The **activation quantizer** applies $\mathcal{Q}^{\mathbf{W}^\top\mathbf{W}}_{\text{reloc}}$ to incoming activations with per-token scales taken from the unquantized $\mathbf{x}$. **Fast Mode** then works on 128-channel blocks (RTN inside a block, relocation only across block boundaries), turning D sequential steps into $D/128$ block GEMMs for a $\sim 100\times$ speedup that retains $\approx 95\%$ of the gain at 4 bits (§4.5). Runtime overhead is one static $D \times D$ FP32 matrix per layer.

3.3 Weight Path: Incoherent Preprocessing and Gaussian Companding

For weights we keep GPTQ relocation as the backbone and compose two operators suggested by Proposition 1. A **block-Hadamard transform** on the input dimension, absorbed into adjacent layers at inference by computational invariance (Ashkboos et al., 2024a), preserves bit-level forward equivalence while making the data Hessian incoherent and tightening the bound. **Gaussian companding** then replaces the uniform absmax lattice: orthogonalized weights are approximately Gaussian, for which uniform grids are MSE-suboptimal, so we use deterministic Lloyd–Max levels (Lloyd, 1982; Max, 1960) fitted to $\mathcal{N}(0,1)$ through the Panter–Dite density $g(x) \propto p(x)^{1/3}$ (Panter & Dite, 1951),

worth $+1.68\,\mathrm{dB}$ SNR for one static 1D lookup table. Both are tuning-free, and together they cut standalone W4 degradation relative to FP16 from $+0.316$ to $+0.102$ PPL, a 68% relative reduction (§4.5).

3.4 Rotation Axis: Composing Mean-Centering and Per-Head Hadamard

Quantization error in key (K) caches decomposes into two mutually orthogonal components: (i) a static DC offset, eliminated by subtracting a calibration-derived channel-mean vector $\boldsymbol{\mu} \in \mathbb{R}^D$, and (ii) a residual dynamic component, dispersed uniformly by an orthonormal per-head Hadamard transform $\mathbf{H}_d$ ($d \in \{64, 128\}$). The composed operator is defined as:

$$\hat{\mathbf{k}} = \boldsymbol{\mu} + \mathbf{H}_d^\top \mathcal{Q}\left(\mathbf{H}_d(\mathbf{k} - \boldsymbol{\mu})\right). \tag{3}$$

The two target disjoint error modes (§4.6): rotation alone has its dynamic range inflated by the static channel biases, and centering alone leaves the dynamic outliers. Granularities are aligned per-token and per-head across all baselines.

3.5 Temporal Axis: Matched Σ-Δ Feedback for Value Caches and Deliberate Pruning

Because value (V) cache representations are aggregated temporally via attention-weighted averaging, first-order temporal error feedback of the form:

$$\mathbf{q}_t = \mathcal{Q}(\mathbf{v}_t + \lambda \mathbf{e}_{t-1}) \tag{4}$$

can in principle push residuals into high-frequency bands that the low-pass character of attention weights attenuates. For an attention row with second-order statistics P_0 and P_1, the distortion from equation 4 is $(1 + \lambda^2)P_0 - 2\lambda P_1$ once a boundary term is dropped, minimized at $\lambda^\star = P_1/P_0 = \rho$, the lag-1 autocorrelation, for a reduction factor $(1 - \rho^2)$; Proposition 3 carries that term and only tightens the bound (Appendix A.3).

With per-head gains $\lambda_h = \rho_h$ the measured improvements track that prediction closely, but in autoregressive decoders $\bar{\rho} \approx 0.05$, so the factor is at best $1 - \bar{\rho}^2 \approx 0.9975$. **We therefore prune the temporal axis from the serving pipeline entirely.** This is the framework working as intended rather than a shortcoming: reading equation 1 *a priori* exposes axes with no practical utility. At 2 bits the value-cache bottleneck moves from temporal dynamics to per-head geometry (§4.6).

3.6 Relocation-Aware Transformation Learning: ROTA-STE

SpinQuant, OSTQuant and FlatQuant optimize equivalent rotations under a surrogate of **memoryless uniform rounding**. If the deployed quantizer relocates error, the training objective must instead reflect the forward dynamics of $\mathcal{Q}_{\mathrm{reloc}}$. **ROTA-STE** optimizes an orthogonal Kronecker transform $\mathbf{R}$ through that exact path:

$$\min_{\mathbf{R}} \mathbb{E}\left[\left\|\mathbf{W}\left(\mathbf{x} - \mathbf{R}^\top \mathcal{Q}_{\mathrm{reloc}}^{\mathbf{RHR}^\top}(\mathbf{R}\mathbf{x})\right)\right\|_2^2\right]. \tag{5}$$

Gradients pass the rounding step via the straight-through estimator (Bengio et al., 2013), and the compensation factor $\mathbf{U} = \mathrm{chol}((\mathbf{R}\mathbf{H}\mathbf{R}^\top)^{-1})$ is updated by alternating minimization. Transformations learned under the memoryless surrogate can fall *below* a static random Hadamard at 3 bits, whereas equation 5 holds up at every precision (§4.5). Applied only to the two most sensitive projections, `o_proj` and `down_proj`, it takes `W4A4KV4` from 6.050 to 5.956.

4 Experiments

We ask whether matched error relocation beats state-of-the-art transformation learning, carries over to weight-only settings, preserves long-context reasoning, and genuinely *relocates* error in a geometric sense.

Table 1: Joint weight-activation-cache quantization. Perplexity is evaluated on WikiText-2 and accuracy represents the macro-average across nine zero-shot tasks; the best result is shown in **bold** and the second-best is underlined. Controlled and uncontrolled variables across columns are detailed in Appendix C.4.

		LLaMA-2 7B		LLaMA-2 13B		LLaMA-3 8B		LLaMA-3.2 1B		LLaMA-3.2 3B		LLaMA-2 70B	
Bits	**Method**	PPL ↓	Acc ↑	PPL ↓	Acc ↑	PPL ↓	Acc ↑	PPL ↓	Acc ↑	PPL ↓	Acc ↑	PPL ↓	Acc ↑
16-16-16	FP16†	5.47	65.08	4.88	67.77	6.13	68.05	9.76	55.75	7.81	63.44	3.32	71.59
W4A4KV4	QuaRot†	6.12	62.03	5.37	65.25	7.82	62.90	14.59	46.81	9.99	57.22	3.80	70.30
	SpinQuant†	5.98	61.60	5.30	65.58	7.50	64.53	13.52	48.35	9.46	56.78	3.61	70.57
	OSTQuant†	5.95	62.32	5.28	65.41	7.42	64.35	13.23	48.80	9.27	58.67	3.59	70.84
	FlatQuant†	6.08	61.97	5.32	65.04	7.73	62.72	13.63	48.20	9.57	58.06	–	–
	ReSpinQuant†	5.74	62.49	5.13	65.82	7.24	64.65	13.03	49.57	9.06	58.84	–	–
	OptQuant	**5.72**	**63.76**	**5.07**	**66.68**	**6.90**	**67.09**	**11.73**	**51.96**	**8.60**	**61.13**	**3.48**	**71.32**
W3A3KV3	QuaRot†	19.26	40.94	12.78	43.67	98.04	33.95	812.5	32.07	172.3	32.94	–	–
	SpinQuant†	9.69	49.15	7.36	55.10	15.07	48.77	69.70	35.02	21.05	42.16	–	–
	OSTQuant†	9.53	48.44	7.10	56.36	14.29	47.34	59.71	35.71	21.27	41.16	–	–
	FlatQuant†	13.32	45.43	9.27	50.95	133.5	33.81	543.7	32.41	84.47	34.13	–	–
	ReSpinQuant†	8.57	50.14	6.91	56.58	13.09	50.74	49.90	36.65	18.61	43.63	–	–
	OptQuant	**6.93**	**57.86**	**5.99**	**62.76**	**10.68**	**55.45**	**37.24**	**39.02**	**14.80**	**49.04**	**4.03**‡	**69.19**‡

4.1 Experimental Setup

We evaluate LLaMA-2 (Touvron et al., 2023) (7B, 13B), LLaMA-3 (Grattafiori et al., 2024) (8B) and LLaMA-3.2 (1B, 3B); other architectures are in Appendix C.12. Perplexity uses the full WikiText-2 (Merity et al., 2017) and C4 (Raffel et al., 2020) test splits at sequence length 2048 (standard GPTQ protocol for C4). Zero-shot accuracy uses `lm-evaluation-harness` (Gao et al., 2023) v0.4.12 over nine benchmarks (BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2020), ARC-e/ARC-c (Clark et al., 2018), OBQA (Mihaylov et al., 2018), LAMBADA (Paperno et al., 2016)), macro-averaged following ReSpinQuant. Weights are symmetric; activations and KV caches asymmetric, per-token and per-head. Calibration draws 32 sequences (64k tokens) from WikiText-2 train; budget and domain-shift sensitivity is ablated in §4.5. We report the tuning-free configuration and the ROTA-STE variant. Baselines are RTN, GPTQ (Frantar et al., 2023), QuaRot (Ashkboos et al., 2024b), SpinQuant (Liu et al., 2025), OSTQuant (Hu et al., 2025), FlatQuant (Sun et al., 2025) and ReSpinQuant (Kim et al., 2026). Rows marked † are cited from ReSpinQuant's unified H100 benchmark, supplemented by our own official-code reproductions of QuaRot and SpinQuant; our FP16 references match cited baselines within ± 0.01 on all five models (Appendix B). Beyond the results reported here, the appendix documents five further studies: a bitwidth ladder from 4 to 2 bits across five models (Appendix C.11), an asymmetric $K \times V$ cache grid that isolates which cache binds first (Appendix C), architectural breadth across other model families (Appendix C.12), serving-time runtime and memory profiling (Appendix C.13), and a batch-size control showing that the sole non-deterministic factor in our harness moves scores by up to 0.57 points, which bounds the margins we are willing to call differences (Appendix B).

4.2 Main Results: Relocation Surpasses Learned Rotations

Table 1 reports `W4A4KV4` and `W3A3KV3`. We cross-referenced the original papers to confirm that every cited entry compresses its KV cache, and label cache bitwidths in both blocks; that verification, the uncontrolled cache granularity in the cited literature, and the gaps between our reproductions and cited numbers are in Appendix C.14.

OptQuant is best in all 22 cells where a rankable baseline exists (Appendix C.14 lists what the cited rows leave uncontrolled). The tuning-free variant alone already beats every transformation-learning baseline: on LLaMA-2-7B it reaches 5.85 against QuaRot (6.12), FlatQuant (6.08), SpinQuant (5.98) and OSTQuant (5.95), all of which *train* their rotations; ROTA-STE gives 5.72 (5.714 ± 0.007, three seeds). The margin widens as precision and capacity fall: on 7B the gap to ReSpinQuant grows from -0.02 at `W4A4KV4` to -1.64 at `W3A3KV3`, and reaches -12.7 on 1B, exactly as §2 predicts: tighter budgets inflate total error, so placement dominates. Accuracy gains are larger still, $+7.7$ points on LLaMA-2-7B at `W3A3KV3`, and per-benchmark decomposition (Table 8) shows no

Table 2: Weight-only quantization under the OSAQ grid on LLaMA-3-8B, per task rather than as a single average. Best entries per block are **bold**; our rows are shaded. † cites OSAQ, and `-cd` is their coordinate-descent variant. Both FP16 baselines are listed because the two harnesses score the same checkpoint differently (70.8 vs. our 71.4), so comparisons hold only within a block. Perplexities come from Table 7. Raw C4 values are not comparable across studies, our FP16 C4 reference being 0.27 to 0.29 higher than OSAQ's (Table 9); read that column as degradation from each side's own baseline (9.47 for us, 9.20 for OSAQ). OSAQ's perplexity and accuracy grids cover slightly different methods, so `OSAQ+QuIP` at `W4A16` has an accuracy but no published perplexity. OSAQ also reports LLaMA-3-70B; we withhold that column because our per-channel symmetric pipeline returns chance-level accuracy on that checkpoint (§5.1).

		LLaMA-3-8B						
Setting	**Method**	**Wiki2 ↓**	**C4 ↓**	PIQA	ARC-e	ARC-c	WinoG.	**Avg**
–	FP16†	6.10	9.20	79.9	80.1	50.4	72.8	70.8
–	FP16 (ours)	6.14	9.47	80.5	77.6	53.9	73.5	71.4
`W4A16`	QuIP†	6.50	11.10	78.2	78.2	47.4	73.2	69.2
	OSAQ+QuIP†	–	–	78.8	**78.9**	48.0	73.1	69.7
	OptQuant	**6.40**	**10.12**	**80.0**	77.9	**51.7**	**76.1**	**71.4**
`W3A16-g128`	AWQ†	8.20	11.60	77.7	74.0	43.2	72.1	66.8
	OSAQ+AWQ†	7.96	**11.40**	**78.2**	**75.1**	43.9	**72.4**	67.4
	GPTQ†	8.20	13.70	74.9	70.5	37.7	71.1	63.6
	OSAQ+GPTQ†	7.98	13.40	76.4	72.2	39.7	72.3	65.2
	OptQuant	**7.10**	11.99	78.0	**75.1**	**49.3**	72.1	**68.6**
`W3A16`	QuIP†	7.50	11.30	76.8	72.9	41.0	72.5	65.8
	OSAQ+QuIP†	7.43	**10.80**	77.6	73.5	41.6	**72.7**	66.4
	OptQuant	**7.17**	12.11	**78.5**	**77.6**	**49.8**	71.6	**69.4**
`W2A16-g128`	GPTQ†	210.00	41000	53.9	28.8	19.9	50.5	38.3
	OSAQ+GPTQ†	63.80	352.00	58.1	38.6	25.8	54.9	44.4
	OSAQ+GPTQ-cd†	26.10	62.00	64.6	44.2	**37.7**	60.4	51.7
	OptQuant	**12.60**	**28.14**	**67.6**	**55.2**	33.7	**63.5**	**55.0**

outlier carries the average: OptQuant leads all nine tasks, widest where naive rounding collapses first (LAMBADA, HellaSwag). The other three models are in Appendix C.

4.3 Weight-Only Quantization under the OSAQ Benchmark Protocol

To show the framework is not confined to joint W-A-KV quantization, we compare weight-only compression against OSAQ (Li et al., 2026) and QuIP (Chee et al., 2023) under OSAQ's own precision and grouping protocol (`W4` per-channel; `W3`/`W2` at group size 128).

Zero-Shot Downstream Accuracy. Table 2 breaks LLaMA-3-8B out by task rather than as one average. Gains concentrate on ARC-challenge, the hardest of the four: +3.7 over OSAQ+QuIP at `W4A16` (51.7 vs. 48.0), +9.6 over OSAQ+GPTQ at `W3A16`-g128 (49.3 vs. 39.7), +8.2 over OSAQ+QuIP at channel-wise `W3A16` (49.8 vs. 41.6), while PIQA and ARC-easy move under a point; `W4A16` is lossless here (71.4 vs. our 71.4 FP16). Two cells go the other way: at channel-wise `W3A16` we lead three tasks but trail on WinoGrande (71.6 vs. 72.7), and at `W2A16`-g128 we lead the average by 3.3 (55.0 vs. 51.7) while their coordinate-descent variant takes ARC-challenge (37.7 vs. 33.7).

Perplexity Dynamics and Cross-Corpus Generalization. Weight-only perplexities are in Appendix C.7 (Table 7). On WikiText-2 OptQuant leads 15 of 16 cells, the exception being channel-wise `W3A16` on LLaMA-2-7B (6.37 for OSAQ+QuIP vs. 6.44). The gap is widest at 2 bits, 1.1× to 2.1× over each column's strongest baseline (MagR-cd on 2-7B/2-70B, OSAQ+GPTQ-cd on 2-

13B/3-8B: $9.94 \rightarrow 8.51$ on 7B, $7.60 \rightarrow 6.62$ on 13B, $26.10 \rightarrow 12.60$ on 8B, $5.52 \rightarrow 4.74$ on 70B) and $2.0\times$ to $5.1\times$ over OSAQ+GPTQ. On 70B we take all four configurations (3.39, 3.61, 3.62, 4.74 vs. baseline bests 3.47, 3.75, 3.81, 5.52), with `W4A16` near-lossless ($3.318 \rightarrow 3.386$), the same low-bit trend as Table 1, where falling precision lets the incoherence–relocation–companding stack compound (§3.3).

On C4 our raw perplexities trail the OSAQ family at `W4` and `W3`, but the two studies' *FP16* baselines disagree: within 0.036 on WikiText-2, ours are 0.27 to 0.29 higher on C4, a penalty incurred before a weight is rounded. Read as degradation from each side's own FP16 baseline, the 4-bit cells flip to us and the 2-bit cells lead widely; only `W3A16`-g128 stays behind, which we attribute to OSAQ's asymmetric grid. Verification and cell-by-cell numbers are in Appendix C.9; the full five-model ladder is in Appendix C.7.

4.4 Long-Context and Complex Reasoning: Beyond Standard Perplexity

With 3-bit KV caches we beat TurboQuant (Zandieh et al., 2025) on all seven LongBench tasks (Bai et al., 2024) (Table 3), retaining 94% of FP16 accuracy against 78%, widest on extractive QA where a corrupted key cache costs retrieval. On Needle-In-A-Haystack every method saturates at 0.960 with 4-bit caches (24 of 25 cells); we report this null result explicitly, since dropping non-separating metrics inflates KV-cache claims.

Table 3: LongBench performance evaluated on Llama-3.1-8B-Instruct with 3-bit KV caches.

Task	FP16	TurboQ.	OptQuant
NarrativeQA (F1)	20.75	13.30	**18.67**
Qasper (F1)	14.79	12.52	**14.61**
MultiFieldQA-en (F1)	27.20	21.73	**25.84**
HotpotQA (F1)	16.90	12.98	**14.69**
2WikiMQA (F1)	16.40	13.30	**15.85**
GovReport (R-L)	21.01	18.04	**20.48**
MultiNews (R-L)	14.72	10.88	**14.22**
Mean (Retention)	18.82	14.68 (78%)	**17.77 (94%)**

On multi-step reasoning (Table 4) naive 4-bit RTN collapses, falling below chance (25%) on MMLU (Hendrycks et al., 2021) and to 0.0% on GSM8K (Cobbe et al., 2021), while our pipeline keeps 91% of FP16 on MMLU. (*GSM8K under simulated quantization needs sequential relocation per generated token, $\approx$1090 s per problem, so MMLU log-likelihood is our primary check of reasoning retention.)

4.5 Ablation Studies and Controlled Baselines

Every operator in the ladder contributes monotonically. Companding without a preceding rotation *hurts* by $+0.332$ PPL, confirming the Gaussian assumption holds only once incoherence is applied; rotation alone gives -0.308, and a structured Hadamard adds only -0.017 over a random orthogonal matrix, as Proposition 1 predicts (Appendix C.3).

Table 4: Full `W4A4KV4` reasoning performance on LLaMA-2-7B.

Benchmark	FP16	RTN	OptQuant
MMLU (0-shot)	41.81	23.28	**38.11**
GSM8K (5-shot)	13.8	**0.0**	n/a*

4.6 Analytical Dissection: Is Error Genuinely Relocated?

Inspecting the residuals (Appendix C.1), relocation *raises* raw error while lowering downstream distortion in all 273 modules of five models (Appendix Figure 2): error is steered, not removed. The downstream eigenbasis separates the levers. Rotation then rounding cuts distortion to $0.105\times$ RTN but leaves the top-10% sensitive subspace's share near isotropic ($12.4\% \rightarrow 10.0\%$); relocation cuts distortion only to $0.687\times$ yet drives that share to 1.6%; together, $0.024\times$ and 0.7%. The axes being independent, rotation-only methods settle on a common plateau.

These gains are foreseeable before a tensor is rounded: the AM/GM coding gain from static weights alone correlates with measured relocation gain at $r = 0.92$ to 0.98 on every projection but `down_proj`, where SwiGLU's heavy tails break the Gaussian assumption (predicted 11.5–13.8 dB vs. measured 1.7–4.5 dB), precisely why `down_proj` and `o_proj` go to ROTA-STE.

Even a poorly chosen statistic yields an unambiguous diagnosis. Sweeping K and V precision independently (Table 11), the two are interchangeable at 3–4 bits (K4V3 vs. K3V4 differ by 0.022, within seed variance) but diverge at 2, where starving V costs a further 4.9 to 5.3 PPL, exposing the

value cache as the bottleneck. Kurtosis calls that cache well-behaved ($3.6 \rightarrow 3.5$ after rotation), yet per-head crest factors reach a 99th percentile of 10.19 beside a near-Gaussian median head (2.95), and the 21 heads above 6 force 82% of the 4-bit cache to zero. Mean centering with a per-head Hadamard removes every head above 4, yet changes nothing above 2 bits (5.8186 vs. 5.8194): an operator pays only where its axis binds. At 2 bits V needs **incoherence alone**, a third instance of Proposition 1; above 2 bits it needs nothing, so the deployed configuration keeps per-head rounding (§5.2).

5 Discussion and Limitations

Claiming each axis's utility is estimable *a priori* obliges us to report where that estimate settles the question and where it leaves one open.

5.1 Methodological Boundaries: 3-Bit Grouped Weight-Only Quantization

Read as degradation from each side's own FP16 baseline, the only rigorous reading the baseline protocol permits (Appendix C.9), one cell still trails on all four models: 3-bit grouped weight-only, `W3A16`-g128. Two hypotheses fit and our evidence cannot separate them: OSAQ's weight quantizer is asymmetric with explicit zero-points where ours is symmetric, and asymmetric grids are more expressive at intermediate bit-widths; and relocation steers under the metric induced by calibration samples, making it the more distribution-coupled of the two exactly where total error is small. At two bits, where placement dominates, we lead on both corpora (§4.2, §4.3); separating the two needs a rerun with an asymmetric quantizer, which we have not done.

5.2 Scope of A Priori Estimation: Value Caches and the Choice of Diagnostic Statistics

The value cache marks the boundary of what *a priori* estimation settles on its own. Ruling out the temporal axis via $\bar{\rho} \approx 0.05$ was correct, but that verdict speaks only to that axis and not to what V needs elsewhere, so the initial pipeline left it at naive rounding. Only after the cache-bitwidth sweep (§4.6) isolated V did mean removal with a head-wise Hadamard recover 37% to 43%. The statistic we selected in advance (kurtosis) does not capture the penalty absmax rounding actually incurs (crest factor). **The framework fixes the optimization question; selecting the summary statistic that answers it remains a modeling choice.** One observation remains open: mean subtraction on V alone diverges (2325.67 PPL), which none of our three offline diagnostics anticipated. The temporal axis is also rejected only for autoregressive decoder attention: where attention flattens, Proposition 3's gain revives.

6 Conclusion

A quantizer's real operational degree of freedom lies not in how much error it generates but in where it places that error. We formalized this as matched error relocation and realized it in **OptQuant**: for each tensor, identify the consuming operator, read the eigenspectrum of the induced metric, and deploy closed-form operators only where that metric forecasts a gain. The formulation subsumes several established techniques as special cases, yields two new operations, and gives the strongest `W4A4KV4` results we are aware of, the tuning-free configuration alone already beating methods that learn their transformations (Appendix C.14 states the caveat on cited baselines).

The framework's defining characteristic is negative: the estimation that selects viable axes also rejects futile ones, and temporal noise shaping was dropped on the measured $\bar{\rho} \approx 0.05$, not by ablation. Even where the estimate left a gap, that gap sat at well-defined coordinates inside the theory, which is what we expect to generalize.

AI Use Statement

In this work, we used generative AI tools for none of the tasks with required disclosure. We have not used generative AI tools for research ideation, core methodology design, data collection, or

experimental analysis, and code generation or synthetic data tasks are not applicable to this work. Additionally, we used generative AI tools for grammar correction and language polishing. We have reviewed all AI-assisted work. Specifically, all drafts were initially authored by the researchers, and AI-suggested grammatical modifications were selectively adopted only after human verification to maintain technical clarity and accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. SliceGPT: Compress large language models by deleting rows and columns. In *International Conference on Learning Representations (ICLR)*, 2024a.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b.

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)*, 2024.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*, 2020.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. QuIP: 2-bit quantization of large language models with guarantees. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *NAACL-HLT*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, et al. A framework for few-shot language model evaluation, 2023. Software, EleutherAI.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

Xing Hu, Yuan Cheng, Dawei Yang, Zhihang Xu, Zhe Yuan, Jiangyong Yu, Chen Xu, Zhe Jiang, and Sifan Zhou. OSTQuant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. In *International Conference on Learning Representations (ICLR)*, 2025.

Suyoung Kim, Sunghyun Wee, Hyeonjin Kim, Kyomin Hwang, Hyunho Lee, and Nojun Kwak. ReSpinQuant: Efficient layer-wise LLM quantization via subspace residual rotation approximation. *arXiv preprint arXiv:2604.11080*, 2026.

Zhikai Li, Zhen Dong, Xuewen Liu, Jing Zhang, and Qingyi Gu. OSAQ: Outlier self-absorption for accurate low-bit LLM quantization. *arXiv preprint arXiv:2605.04738*, 2026.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems (MLSys)*, 2024.

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. In *International Conference on Learning Representations (ICLR)*, 2025.

Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28 (2):129–137, 1982.

Joel Max. Quantizing for minimum distortion. *IRE Transactions on Information Theory*, 6(1):7–12, 1960.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

P. F. Panter and W. Dite. Quantization distortion in pulse-count modulation with nonuniform spacing of levels. *Proceedings of the IRE*, 39(1):44–48, 1951.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *AAAI Conference on Artificial Intelligence*, 2020.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. FlatQuant: Flatness matters for LLM quantization. In *International Conference on Machine Learning (ICML)*, 2025.

Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning (ICML)*, 2023.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. TurboQuant: Online vector quantization with near-optimal distortion rate. *arXiv preprint arXiv:2504.19874*, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

A PROOFS

Throughout this section, let $\mathbf{H} \succ 0$ (guaranteed by the ridge damping term δ in Eq. equation 1), $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$, and the metric distortion be given by $\mathcal{D}(\mathbf{e}) = \mathbf{e}^\top \mathbf{H}\mathbf{e}$. The symbol D denotes the input dimension.

A.1 PROOF OF PROPOSITION 1

Proposition 1 (Relocation Gain and Its Theoretical Ceiling)**.** *Let $\mathbf{H} = \mathbf{N}^\top \mathbf{\Lambda} \mathbf{N}$ denote the triangular factorization induced by the quantization ordering, where $\mathbf{N}$ is unit lower-triangular and $\mathbf{\Lambda} = \mathrm{diag}(d_1, \ldots, d_D)$. Because $\mathbf{N}^\top$ is unit* upper*-triangular, this represents a $UDU^\top$ decomposition; that is, d_j are the $LDL^\top$ pivots under reversed coordinate ordering. Suppose the scalar rounding residuals δ_j at each step are zero-mean, share a common variance σ^2, and are uncorrelated across coordinates. Then:*

$$\mathbb{E}\big[\mathcal{D}_{\mathrm{reloc}}\big] = \sigma^2 \sum_{j=1}^{D} d_j, \qquad \mathbb{E}\big[\mathcal{D}_{\mathrm{RTN}}\big] = \sigma^2 \,\mathrm{tr}(\mathbf{H}).$$

Because $d_j \le \mathbf{H}_{jj}$ for all j, relocation is guaranteed never to increase expected metric distortion. Furthermore, the achievable gain $G(\mathbf{H}) = \mathrm{tr}(\mathbf{H}) / \sum_j d_j$ satisfies:

$$G(\mathbf{H}) \;\le\; \frac{\mathrm{tr}(\mathbf{H})/D}{\det(\mathbf{H})^{1/D}},$$

which is the arithmetic-to-geometric mean ratio of the eigenvalues of $\mathbf{H}$. Equality holds if and only if all pivots are identical. An orthogonal coordinate transformation $\mathbf{H} \mapsto \mathbf{T}\mathbf{H}\mathbf{T}^\top$ preserves both $\mathrm{tr}(\mathbf{H})$ and $\det(\mathbf{H})$, and therefore leaves this ceiling invariant.

Proof. As in Eq. equation 2, let $\mathbf{H}^{-1} = \mathbf{U}^\top \mathbf{U}$ with upper-triangular $\mathbf{U}$, and define $\mathbf{M} = \mathbf{U}^{-\top}$. Then $\mathbf{M}$ is lower-triangular and $\mathbf{H} = \mathbf{M}^\top \mathbf{M}$. Consequently:

$$\mathbf{e}^\top \mathbf{H}\mathbf{e} = \|\mathbf{M}\mathbf{e}\|_2^2 = \sum_{j=1}^{D} \mathbf{M}_{jj}^2 \Big(e_j + \sum_{k<j} \tfrac{\mathbf{M}_{jk}}{\mathbf{M}_{jj}} e_k \Big)^2, \qquad d_j := \mathbf{M}_{jj}^2 = 1/\mathbf{U}_{jj}^2.$$

Factoring $\mathbf{M} = \mathbf{\Lambda}^{1/2}\mathbf{N}$ into diagonal and unit lower-triangular components recovers the stated decomposition $\mathbf{H} = \mathbf{N}^\top \mathbf{\Lambda} \mathbf{N}$, with $\prod_j d_j = \det(\mathbf{U})^{-2} = \det(\mathbf{H})$. At step j, because the sequential update in Eq. equation 2 has absorbed the accumulated contributions of $e_1, \ldots, e_{j-1}$ into the pending coordinates, the bracketed term reduces strictly to the scalar rounding residual δ_j determined at that step. Thus, $\mathcal{D}_{\mathrm{reloc}} = \sum_j d_j \delta_j^2$, taking expectations yields the first identity. Memoryless rounding does not compensate $\mathbf{e}$, yielding $\mathbb{E}[\mathbf{e}^\top \mathbf{H}\mathbf{e}] = \sigma^2 \,\mathrm{tr}(\mathbf{H})$.

Each pivot is a Schur complement $d_j = \mathbf{H}_{jj} - \mathbf{h}_j^\top \mathbf{S}_j^{-1} \mathbf{h}_j$ with $\mathbf{S}_j \succ 0$, which implies $d_j \le \mathbf{H}_{jj}$ and hence $\sum_j d_j \le \mathrm{tr}(\mathbf{H})$. For the upper bound, the AM–GM inequality dictates that $\sum_j d_j \ge D\big(\prod_j d_j\big)^{1/D} = D \det(\mathbf{H})^{1/D}$, with equality holding if and only if all d_j are identical. Substituting this into $G(\mathbf{H})$ yields the claimed bound. Finally, for any orthogonal $\mathbf{T}$, $\mathrm{tr}(\mathbf{T}\mathbf{H}\mathbf{T}^\top) = \mathrm{tr}(\mathbf{H})$ and $\det(\mathbf{T}\mathbf{H}\mathbf{T}^\top) = \det(\mathbf{H})$, leaving the right-hand side invariant. □

Remark (What Transformations Actually Gain). Because the theoretical ceiling is invariant under orthogonal $\mathbf{T}$, transformations cannot increase the attainable headroom. Their sole effect is to equalize the pivots, driving realized gains toward the theoretical ceiling. This clarifies the precise operational mechanism of incoherent preprocessing: it implies that the specific choice among incoherent transformations is largely inconsequential. This expectation is confirmed empirically in §4.5, where structured Hadamard matrices provide merely -0.017 PPL over random orthogonal matrices, accounting for approximately 5% of the overall gain. We state this as an empirical observation rather than a theorem; a formal quantitative bound connecting incoherence parameters directly to pivot concentration remains unproven.

Remark (Raw Energy vs. Metric Distortion). Proposition 1 bounds $\mathbf{e}^\top \mathbf{H}\mathbf{e}$, not the raw residual norm $\|\mathbf{e}\|_2^2$. These represent distinct objectives that need not move in tandem. §4.6 documents numerous modules where relocation actively inflates $\|\mathbf{e}\|_2^2$ while halving $\|\mathbf{W}\mathbf{e}\|_2^2$.

A.2 Proof of Proposition 2

Proposition 2 (Local Exactness of the Activation Metric)**.** *For activation vectors consumed by a single linear layer* $\mathbf{y} = \mathbf{W}\mathbf{x}$*, the distortion* at the output of that layer *is governed exactly by the quadratic form with* $\mathbf{H}_0 = \mathbf{W}^\top \mathbf{W}$*. This matrix* $\mathbf{H}_0$ *is determined by static weights, is identical across all tokens, and requires zero calibration data. The metric deployed in practice is the ridged matrix* $\mathbf{H}$ *in Eq. equation 1. When* $\mathbf{W}$ *has fewer rows than columns (as in GQA key/value projections and* `down_proj`*),* $\mathbf{H}_0$ *is rank-deficient; the ridge is therefore not a numerical convenience, but a mathematical necessity ensuring that* $\mathbf{H}^{-1}$ *exists. Incorporating the ridge, the Cholesky factor of* $\mathbf{H}^{-1}$ *is computed only once per layer offline, and the recurrence in Eq. equation 2 vectorizes across the token dimension.*

Proof. The identity $\|\mathbf{W}\mathbf{x} - \mathbf{W}\hat{\mathbf{x}}\|_2^2 = (\mathbf{x} - \hat{\mathbf{x}})^\top \mathbf{W}^\top \mathbf{W}(\mathbf{x} - \hat{\mathbf{x}})$ is an exact algebraic equality rather than an approximation. Because $\mathbf{W}$ is token-independent, neither $\mathbf{H}$, its inverse, nor its derived triangular factor depends on the input tokens. The per-token computation consists of the recurrence in Eq. equation 2 with token-independent coefficients; this is a sequence of rank-1 updates applied identically across all columns, and is therefore fully vectorizable. The corresponding objective for weight quantization is $\|(\mathbf{W} - \hat{\mathbf{W}})\mathbf{X}\|_F^2 = \mathrm{tr}\big((\mathbf{W} - \hat{\mathbf{W}})\mathbf{X}\mathbf{X}^\top(\mathbf{W} - \hat{\mathbf{W}})^\top\big)$, where the metric $\mathbb{E}[\mathbf{X}\mathbf{X}^\top]$ is an empirical property of the input data that must be estimated from samples. □

Scope of Applicability. This proposition applies strictly to the output of the layer consuming the tensor, rather than to the end-to-end task loss. Residual connections, layer normalizations, attention non-linearities, and input-dependent dynamics all interleave this layer and the final network output. The coupling between this local objective and task perplexity is treated as an empirical premise and validated systematically in §4.6. There, our predictions hold with correlations between $r = 0.92$ and 0.98 across all projections except `down_proj`, which represents the sole pathway separated from its consumer by SwiGLU.

A.3 Proof of Proposition 3

Proposition 3 (Matched Feedback Gain)**.** *Let* $\mathbf{a}$ *denote an attention row with second-order statistics* $P_0 = \mathbb{E}\big[\sum_s a_s^2\big]$*,* $P_1 = \mathbb{E}\big[\sum_s a_s a_{s+1}\big]$*, and* $\kappa = \mathbb{E}[a_1^2]/P_0$*. Suppose the value cache is quantized with first-order error feedback as in Eq. equation 4, starting from* $e_0 = 0$*. If the step-wise residuals are zero-mean, share variance* σ^2*, are mutually uncorrelated, and are independent of* $\mathbf{a}$*, the distortion at the attention output equals* $\sigma^2\big[P_0 - 2\lambda P_1 + \lambda^2(1-\kappa)P_0\big]$*. This distortion attains its minimum at* $\lambda^\star = P_1/\big((1-\kappa)P_0\big)$*, where the distortion is reduced by a factor of* $1 - \rho^2/(1-\kappa)$ *relative to the uncompensated case (*$\lambda = 0$*). Here,* $\rho = P_1/P_0$ *represents the normalized lag-1 autocorrelation of the attention weights. The boundary parameter* κ *arises from the boundary condition* $a_{T+1} := 0$*; neglecting this boundary term recovers the familiar expressions* $\lambda^\star = \rho$ *and* $(1 - \rho^2)$*.*

Proof. Writing the residual established at step t as $e_t = (\mathbf{v}_t + \lambda e_{t-1}) - q_t$, the reconstruction error of the cached value is $r_t = q_t - \mathbf{v}_t = \lambda e_{t-1} - e_t$. Because attention outputs $\sum_s a_s q_s$, the error in the aggregated vector is $\sum_s a_s r_s$. Regrouping coefficients for each e_s gives:

$$\sum_s a_s r_s = \sum_s e_s\,(\lambda a_{s+1} - a_s), \qquad a_{T+1} := 0.$$

Because e_s are zero-mean, share variance σ^2, and are mutually uncorrelated:

$$\mathbb{E}\Big|\sum_s a_s r_s\Big|^2 = \sigma^2 \sum_s (a_s - \lambda a_{s+1})^2 = \sigma^2\big[P_0 - 2\lambda P_1 + \lambda^2(1-\kappa)P_0\big],$$

since truncating the final index yields $\sum_s a_{s+1}^2 = P_0 - a_1^2 = (1-\kappa)P_0$ in expectation. Because $(1-\kappa)P_0 > 0$, this expression is quadratic and strictly convex in λ. Setting the derivative $2\lambda(1-\kappa)P_0 - 2P_1$ to zero yields $\lambda^\star = P_1/\big((1-\kappa)P_0\big)$. Substituting this back leaves $P_0 - P_1^2/\big((1-\kappa)P_0\big)$, which scales by a factor of $1 - \rho^2/(1-\kappa)$ relative to the uncompensated baseline P_0. □

Remark. This bound characterizes the maximum theoretical headroom afforded by this axis, rather than the empirical gain achieved by a specific implementation. Because residuals are assumed to be white noise, it represents an upper bound on gain. Substituting the empirical autocorrelation $\bar{\rho} \approx 0.05$ measured in §3.5, the achievable reduction is at most $\bar{\rho}^2/(1-\kappa)$. Neglecting the boundary term yields 0.25%, and even with an extreme boundary factor $\kappa = 0.5$, the reduction reaches merely $\approx 0.5\%$. This confirms that pruning this axis entirely is justified.

B Reproduction Details

Operator Hyperparameters. Weight path: `gptq_had_gauss`, weight Hessian ridge damping $\delta = 0.05$, activation reordering disabled, group sizes configured as tabulated, and block-Hadamard applied using the largest power-of-two divisor of the input dimension. Activation path: $\delta = 0.01$, activation reordering enabled, Fast Mode operating over 128-channel blocks, with one static $D \times D$ FP32 compensation factor per group sharing an input. Key caches are quantized per-token and per-head, with channel means calibrated over 8 chunks. ROTA-STE: the orthogonal Kronecker transformation $\mathbf{R} = \mathbf{R}_1 \otimes \mathbf{R}_2$ is parameterized via skew-symmetric matrices through the Cayley transform, optimized using Adam for 120 steps at a learning rate of 2×10^{-3} and $\delta = 0.01$, applied exclusively to `o_proj` and `down_proj`. Compensation factors are updated via alternating minimization. Calibration across all settings uses 32 sequences of 2048 tokens drawn from the WikiText-2 training split, except for weight-only benchmarks which follow OSAQ's 128-sample protocol.

Two Categories of Baseline Numbers. The main results table cites published numbers denoted by †. In contrast, the per-task breakdown table (Table 8) contains no cited rows; all entries represent our own direct reproductions executed on our hardware using the authors' official repositories, ensuring that task-level columns are strictly comparable. We never mix cited and re-run figures within a single table.

Evaluation Harness. All entries in Table 8 are evaluated using `lm-evaluation-harness` 0.4.12 across full test splits. Reaching this unified version required non-trivial reconciliation: our codebase requires `transformers` 5, SpinQuant was released pinned to 0.4.5, and QuaRot pins 0.4.0 (a release predating SIQA). Version 0.4.5 cannot serve as a common target because it imports `AutoModelForVision2Seq`, which was removed in `transformers` 5. Version 0.4.12 was the only release compatible across all three environments, and was installed with `--no-deps` to preserve each repository's native `transformers` dependencies. The evaluation harness is not a benign hyperparameter: holding all other variables constant, upgrading SpinQuant from 0.4.5 to 0.4.12 shifted 9-task macro-averages by $+0.38$ and $+0.31$ points, while naive RTN shifted by -1.30 and $+1.38$ points, margins comparable to the algorithmic gaps being evaluated. We therefore unified the harness across baselines by direct empirical measurement rather than assumption.

Batch Size. This pipeline is fully deterministic in all aspects except the evaluation batch size. Re-running a completed configuration with the identical batch size on a different GPU reproduces results down to full machine precision: for example, SIQA on LLaMA-2-7B returned 0.4616171954964176 across both runs, matching to 16 decimal places. Altering the batch size breaks this invariance. Running the identical model at batch size 64 instead of 32 yields 0.4651995905834186 (a 0.36-point shift), and shifts by 0.57 points on LLaMA-3-8B. The quantizer is not the source of this discrepancy: because activation scales are determined per-token, the quantizer cannot perceive batch grouping. The variance arises because `lm-eval` buckets requests dynamically by length; altered groupings change padding patterns, and under `bfloat16`, these numerical differences flip model predictions on examples closest to decision boundaries. Table 5 provides a complete 9-task comparison on LLaMA-3-8B, the single model evaluated exhaustively across both batch sizes. While the macro-average shifts by 0.33 points, individual tasks fluctuate more substantially and bidirectionally (e.g., ARC-e shifts by 1.89 points), largely canceling in aggregate. Consequently, batch size affects task-level breakdowns rather than overall averages. We document this phenomenon to explain why we avoid synthesizing rows from runs with disparate batch sizes, and to emphasize that reproductions fixing models, harnesses, seeds, and hardware will fail to match our exact precision unless the evaluation batch size is also held fixed.

Table 5: Evaluation batch size represents the sole non-deterministic factor in this evaluation pipeline. W4A4KV4 on LLaMA-3-8B is measured exhaustively across both batch sizes using identical model weights, evaluation harness, and random seeds.

Batch	BoolQ	PIQA	SIQA	HellaS.	WinoG.	ARC-e	ARC-c	OBQA	LAMB.	Avg_9
32	80.89	79.11	47.19	77.52	72.30	75.59	51.28	43.2	73.74	66.76
64	81.01	78.40	46.62	77.35	72.61	77.48	52.05	44.8	73.47	67.09

Environment and Hardware. All results were produced with `torch 2.5.1+cu121`, `transformers`, `datasets 5.0.1` and `lm-evaluation-harness 0.4.12` on NVIDIA H200 accelerators. One cell, W3A3KV3 on LLaMA-2-70B in Table 1, was run on NVIDIA B200 with `torch 2.7.0+cu128` because the pinned environment lacks Blackwell kernels; all other dependencies were unchanged. Baseline reproductions use the official QuaRot and SpinQuant repositories (modifications in the paragraph below); WikiText-2 and C4 are loaded from `Salesforce/wikitext` and `allenai/c4`, and `social_i_qa` from the Hub Parquet conversion, since `datasets 5.0.1` rejects the script-based loader.

To quantify the cross-hardware offset we re-ran the single-card LLaMA-2-7B calibration with identical commands on both systems over a $40{,}960$-token window, so only the deltas are meaningful. FP16 reproduced exactly (5.7261); weight-only W4 shifted by -0.0011 and joint W4A4KV4 by $+0.0440$. Repeated runs are bitwise identical, so the offset is systematic rather than random, and it is not attributable to Triton: forcing companding through PyTorch reference kernels reproduces the same three values. In accuracy terms the offset moved SIQA by -0.46 points, and we therefore do not claim algorithmic superiority between B200 and H200 numbers at margins below that threshold. The 70B W3A3KV3 entry has no baseline in the literature, so no comparison in this paper rests on it.

Modifications Required for QuaRot Reproduction. Reproducing QuaRot on our shared harness required five modifications. None were optional, and results cannot be reproduced without them: four concern the environment and do not alter algorithmic behavior. First, `initialize_tasks()` and `tasks.ALL_TASKS`, which were removed after `lm-eval` 0.4.0, were replaced with `TaskManager`. Second, the `gguf_file` keyword passed by modern `lm-eval` to `from_pretrained` was removed because `transformers` 4.36.2 does not accept this argument. Third, `datasets` was upgraded from 2.17.1 to 2.20.0. Fourth, isolated dataset caches were provisioned per run, as caches written by `datasets` 5.x record feature schemas unparseable by 2.x releases.

The fifth modification alters algorithmic execution, and is reported explicitly. QuaRot's token-level key quantization reshapes $\mathbf{K}$ to `hidden_size` and subsequently restores it to (`bsz`, `seq_len`, `num_heads`, `head_dim`). These shapes match only when $\texttt{num_heads} \times \texttt{head_dim} = \texttt{hidden_size}$. This condition holds in Multi-Head Attention but fails in Grouped-Query Attention. LLaMA-3-8B has eight key-value heads of dimension 128 against a hidden size of 4096. This mismatch does not produce a clean error: as long as $\texttt{bsz} \times \texttt{seq_len}$ is a multiple of 4, the reshape succeeds, inadvertently grouping keys across four distinct tokens into a single quantization row without warning. This contradicts the token-level operation described in the authors' code comments. Exceptions occur only in batches with non-divisible token counts. Fixed-length perplexity evaluations at length 2048 never expose this bug, but the variable-length requests generated by evaluation harnesses do. We corrected this by quantizing according to the true physical width of the tensor. This correction substantially strengthens QuaRot on LLaMA-3-8B: perplexity improves from 10.881 to 8.877, and 9-task average accuracy increases by $+6.00$ points, thereby compressing our own performance advantage. Multi-head models remain unaffected, and LLaMA-2-7B reproduces identically under both implementations.

Instruction-Tuned Models. EXAONE is evaluated using its native chat template, and LAMBADA is excluded from its macro-average. Evaluating EXAONE-4.0-1.2B under FP16 across both protocols reveals that the raw text protocol leaves the model at chance performance on five of eight multiple-choice benchmarks, whereas applying the template elevates all eight benchmarks above chance. LAMBADA is omitted because chat templates that terminate assistant turns preclude evaluating exact next-token continuation.

B.1 Methodological Boundaries: 3-Bit Grouped Weight-Only Quantization

When evaluated via degradation relative to respective FP16 baselines (the only rigorous reading permitted by the baseline protocol, as demonstrated in Appendix C.9), a single evaluation cell remains trailing across all four models: 3-bit grouped weight-only compression (`W3A16`-g128). We attribute this residual deficit to unaligned configuration choices rather than the relocation algorithm itself. OSAQ quantizes weights asymmetrically with explicit zero-points whereas our scheme is symmetric; asymmetric quantizers offer superior expressive capacity, and this advantage manifests most prominently in intermediate bit regimes.

Alongside this structural explanation, a second hypothesis is plausible, and current empirical evidence cannot separate them. Because relocation steers residuals under an empirical metric $\mathbf{H}$ induced by the calibration distribution, its effectiveness depends on how closely the evaluation distribution aligns with that metric, whereas OSAQ's null-space suppression relies less on distribution-specific assumptions. In grouped 3-bit regimes, total error magnitude is small and relocation headroom is narrow, allowing differences in distributional robustness to govern performance rankings. In contrast, in 2-bit regimes and joint W-A-KV configurations where error placement dominates, our method leads across both evaluation corpora (§4.2, §4.3). Disentangling these two hypotheses would require re-running the pipeline with an asymmetric weight quantizer, which was not undertaken here. Because the two approaches operate along orthogonal axes, composing them represents a natural subsequent step.

B.2 Scope of A Priori Estimation: Value Caches and the Choice of Diagnostic Statistics

The value cache marks the boundary of what our *a priori* estimation settles on its own. While ruling out the temporal axis via $\bar{\rho} \approx 0.05$ proved correct, that analysis provided no guidance regarding what V required along other axes, leading our initial pipeline to leave V to naive rounding. Only after grid sweeps across cache bitwidths (§4.6) isolated V as the 2-bit bottleneck did we trace the root cause to the heavy tail of per-head crest factors, where compensatory rotations recovered 37% to 43% of the degradation. The underlying lesson is that our a priori chosen statistic (kurtosis) did not reflect the metric actively penalized by absmax rounding (crest factor). **While our framework formalizes the correct optimization question, choosing the exact summary statistic to answer that question remains a human design choice.** An open question remains in this context: applying mean subtraction exclusively to V induces catastrophic divergence, yielding a perplexity of 2325.67. None of our three offline diagnostics (crest factor, reconstruction error, or attention residual bias) anticipated this divergence, and the exact blind spot in our diagnostic toolkit remains to be characterized.

Rejecting the temporal axis was an intended prediction of our framework rather than a shortfall. However, the empirical rationale for this rejection ($\bar{\rho} \approx 0.05$) is not a universal physical constant, but an architectural property of autoregressive decoder attention. In regimes where attention distributions flatten, such as bidirectional encoders or retrieval across wide contexts, ρ can increase substantially, potentially reviving the $(1-\rho^2)$ reduction factor established in Proposition 3. Formalizing the precise conditions under which this axis reactivates remains an open problem.

B.3 System and Scalability Limitations

This subsection expands upon the summary presented at the end of §5.

Constraints of Simulated Quantization. Because our implementation evaluates quantization via simulation, reported memory figures represent analytically derived quantities rather than measured hardware allocations (detailed in §C.13). Furthermore, measuring end-to-end INT4 GEMM speedups lies outside the scope of this work. Instead, we report that the additional runtime overhead introduced by the relocation operator remains empirically within 16.5% of baseline layer GEMM execution under `torch.compile`. We accompany this with a negative engineering result: three custom-written fused Triton kernels did not outperform the compiler-generated code. Generative benchmarks face an identical constraint: because evaluation harnesses invoke the sequential relocation logic repeatedly for every generated token, evaluating GSM8K required ~ 1090 seconds per

problem, rendering complete evaluation computationally prohibitive. Consequently, our claims regarding reasoning retention rely on log-likelihood evaluations on MMLU, where our method retains 91% of FP16 performance. Coupling these operators with production-ready INT4 execution kernels represents the immediate next step for this line of research.

Scaling to Large Models and Architectural Extensions. Scaling to larger model dimensions initially revealed two apparent obstacles, though closer inspection dissolved the first. ROTA-STE optimization includes the $D \times D$ compensation matrices of `o_proj` and `down_proj` within the autograd graph. This induced out-of-memory errors on single-GPU 33B runs, which we initially interpreted as a fundamental scalability limit of the learning-based variant. That interpretation was incorrect: because 70B models cannot be evaluated on a single accelerator under any circumstances, this benchmark measured a constraint that the target deployment never encounters. Once proper multi-device sharding was engaged, the optimization fit within memory and trained reliably. The learned variant reaches 3.4868 on 70B models, compared to 3.4781 for the tuning-free variant. What changes with scale is not whether transformations can be learned, but whether learning them remains worthwhile: while learning yields a 0.14 perplexity gain on 7B (5.714 vs. 5.85), it performs slightly worse on 70B, which is why we report the tuning-free configuration at that scale. Because competing methods learn transformations and report 70B results, whether optimization scales was never the central issue. The second obstacle is real, and applies equally to the tuning-free configuration: the relocation operator maintains a static $D \times D$ compensation factor in memory per quantized module, totaling 365 GB in FP32 across all seven projections of 80 layers on 70B models, where `down_proj` (28672^2 per layer) accounts for the vast majority. Storing factors in FP16 halves this footprint, and we verified that this incurs zero loss in accuracy on LLaMA-2-7B (5.7148 in FP16 vs. our three-seed FP32 baseline of 5.714 ± 0.007, representing a 0.09σ shift; Appendix C).

Maintaining resident compensation factors per module represents a genuine architectural overhead of error relocation that rotation-based frameworks do not incur. Compressing these structures via low-rank or banded parameterizations to sub-quadratic complexity constitutes the most critical engineering challenge left by this work. Consequently, we report only tuning-free configurations for large models (33B and 70B), aligning with our architectural coverage protocol in §C.12. As noted above, this choice is not driven by an inability to scale the learning-based variant, but because learning does not justify its computational cost at that scale. Beyond these models, Mixture-of-Experts (MoE) architectures were not evaluated; the Gemma-2 family was excluded due to an evaluation harness anomaly involving logit soft-capping interactions; and multimodal EXAONE-4.5 checkpoints were evaluated with quantization restricted strictly to their language backbones.

C Additional Experimental Results

C.1 Extended Analytical Dissection

This subsection provides the uncompressed narrative corresponding to the summary in §4.6.

Finally, we analyze the geometric properties of quantization error residuals directly.

(a) Empirical Verification of the Conservation Law. Error relocation increases raw tensor distortion $\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2$ while simultaneously suppressing downstream functional distortion $\|\mathbf{W}(\mathbf{x}-\hat{\mathbf{x}})\|_2^2$. This inverse relationship holds without exception across all 273 modules evaluated across five model architectures (Figure 2), providing direct empirical proof that quantization error is geometrically steered rather than eliminated.

(b) A Priori Prediction of Relocation Gains. Theoretical AM/GM coding gains derived strictly from static weight matrices correlate with measured relocation gains at coefficients between $r = 0.92$ and 0.98 across all projections except `down_proj`. The prediction gap on `down_proj` (predicted gain 11.5 to 13.8 dB vs. measured gain 1.7 to 4.5 dB) arises because heavy-tailed activations downstream of SwiGLU violate the underlying Gaussian assumption. This departure provides an independent theoretical justification for assigning `down_proj` and `o_proj` to ROTA-STE in §3.6. Furthermore, GQA projections (`k_proj`, `v_proj`) exhibit rank deficiency, unlocking approximately twice the theoretical relocation headroom observed in standard MHA layers and explaining why empirical performance gaps expand on GQA architectures.

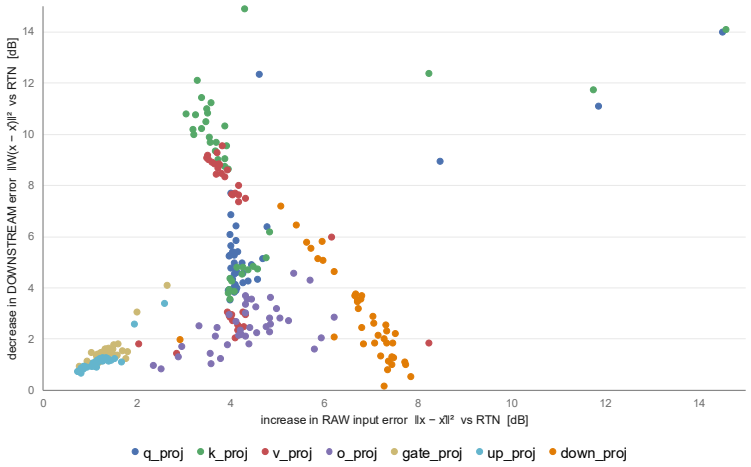


Figure 2: Empirical verification of error conservation. Relocation increases raw input tensor error while suppressing the distortion observed by the downstream layer. This property holds across every single one of the 273 modules evaluated across five model architectures.

(c) Mechanism Decomposition: The Plateau of Pure Rotations. Projecting error onto the downstream eigenbasis demonstrates that rotation followed by rounding suppresses downstream distortion $\|\mathbf{W}(\mathbf{x} - \hat{\mathbf{x}})\|^2$ to $0.105\times$ relative to RTN, yet leaves the proportion of error energy concentrated in the top 10% sensitive subspace near the isotropic baseline ($12.4\% \rightarrow 10.0\%$). Conversely, relocation reduces distortion to $0.687\times$ while suppressing the sensitive subspace proportion down to 1.6%. Combining both levers achieves $0.024\times$ distortion suppression and 0.7% sensitive subspace concentration. Because rotation and relocation operate along independent control axes, methods relying exclusively on rotations (spanning QuaRot to ReSpinQuant) inevitably converge onto a common performance plateau.

(d) Value Cache Diagnosis: Identifying Local Bottlenecks. While aggregate kurtosis suggests that the value cache is well-behaved ($3.6 \rightarrow 3.5$ post-rotation), per-head crest factor analysis reveals that while the median head is Gaussian-like (2.95), the top 1% of extreme heads forms a severe heavy tail reaching a 99th percentile of 10.19. The 21 heads exhibiting crest factors above 6 cause 82% of cached values to clip to zero under 4-bit quantization. Composing mean centering with per-head Hadamard transforms eliminates this tail in sample heads and recovers 37% to 43% of the degradation (Table 11, the centered-and-rotated $V{=}2$ column), with mean subtraction accounting for the vast majority of this recovery. Consequently, what the value cache requires at 2 bits is **incoherence alone**, providing a third independent empirical confirmation of Proposition 1. We report this strictly as a low-bit diagnostic observation rather than as part of our deployment stack: the production configuration quantizes V via per-head RTN (§C.10). Figure 3 shows why aggregate statistics do not reveal this behaviour.

The observation that applying mean subtraction in isolation induces catastrophic divergence (2325.67 PPL) remains an open problem. This mode was not anticipated by any of our three offline diagnostics.

C.2 Figures Supporting Analytical Dissection

The following figures illustrate the empirical measurements discussed in §4.6, showing where error projects within the downstream eigenbasis and quantifying how far the spectrum of $\mathbf{H}$ deviates from isotropy.

C.3 Ablation Studies and Controlled Baselines

Every operator across our ladder contributes monotonic gains (Table 14a), with row L4 serving as a counterfactual control: applying companding without preceding rotations *degrades* perplexity by $+0.332$, confirming that Gaussian density assumptions hold only when incoherent transformations

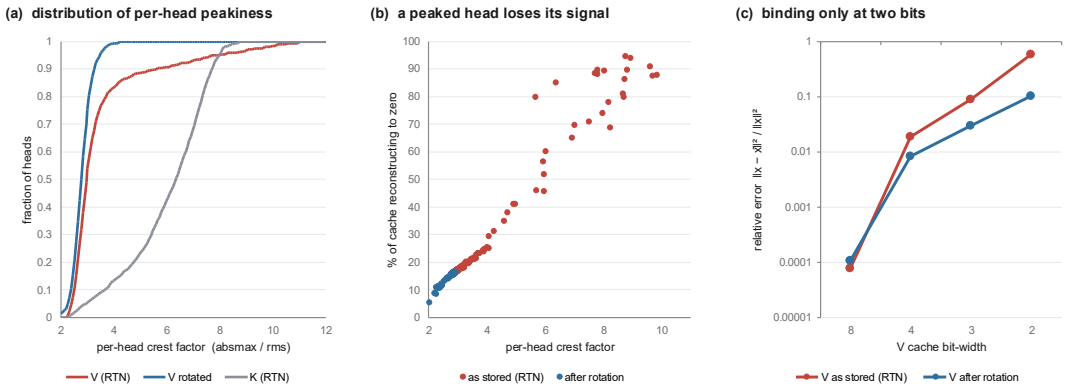


Figure 3: Why the value cache binds at 2 bits, and why kurtosis does not reveal it. **(a)** Because absmax rounding pays a penalty proportional to the maximum value in each head, the governing statistic is the crest factor (absmax divided by RMS) rather than the marginal distribution. The median head in V is Gaussian within sampling noise (2.95), but the extreme tail reaches 10.19, causing head-averaged metrics to characterize V as clean. **(b)** A head whose peak is ten times its RMS allocates its entire discrete budget to that peak. Across a sample of 256 heads at 4 bits, the 21 heads with crest factors above 6 reconstruct 82% of their cache entries to exactly zero, compared to 17% for the 223 heads below 4. Applying per-head Hadamard transforms eliminates all heads with crest factors exceeding 4, eradicating this pathological behavior. **(c)** While this correction reduces reconstruction error across all bitwidths, it impacts perplexity exclusively at 2 bits, because V becomes a binding constraint only below 3 bits.

are actively engaged. Table 14b confirms three findings. First, rotation yields a -0.308 PPL gain, whereas structured Hadamard matrices add merely -0.017 (4.9σ, $\approx 5\%$ of total gain) over random orthogonal matrices, corroborating Proposition 1: **performance is governed by incoherence itself rather than any specific basis**. Second, performance is largely insensitive to group size and calibration domain. Third, quadrupling calibration samples degrades unrotated GPTQ by $+0.132$ while improving rotated GPTQ by -0.037.

C.4 Notes on Primary Tables

Guidance on interpreting Table 1 and Table 2 is provided in their captions; controlled and uncontrolled variables are documented here.

Weight-only quantization under the OSAQ grid is partitioned between the main text and this appendix. Table 2 reports task-level zero-shot accuracy across the two LLaMA-3 models documented by OSAQ. Model-level WikiText-2 and C4 perplexities, alongside four-task zero-shot averages, are tabulated jointly. That table is an abridged summary; omitted baselines are provided in Table 7. Rows marked with † are cited from OSAQ. Because that work reports zero-shot benchmarks only for LLaMA-3-8B and LLaMA-3-70B, accuracy figures are absent for other cited models. GPTQ is listed exclusively from citations: because our error feedback implementation is the relocation operator itself and stripping its Hadamard stage fails to faithfully reproduce reference GPTQ (exhibiting numerical instability absent in canonical GPTQ), we do not report our implementation as generic GPTQ. All evaluated cells (**OptQuant**) were obtained on B200 accelerators at batch size 32; corresponding hardware offsets are quantified in Appendix B. C4 columns cannot be compared directly between studies (Table 9).

Table 1 evaluates six models under joint weight-activation-cache quantization, reporting WikiText-2 perplexity and 9-task zero-shot averages. Baselines for 70B models are cited from QuaRot, SpinQuant, and OSTQuant; missing entries reflect omissions in the original works rather than this study. ReSpinQuant notes that evaluations were limited to models up to 13B on a single accelerator; FlatQuant's 70B results evaluate LLaMA-3-70B rather than this checkpoint; and no cited literature evaluates 70B models under `W3A3KV3`. This table compares transformation-based methods exclusively. RTN and GPTQ are weight-only quantizers, and evaluating them in joint settings requires postulating an activation quantizer never proposed in those works; they are therefore deferred

to weight-only comparisons (Table 7). One caveat applies to our results: our 70B entry uses a tuning-free configuration, whereas smaller models utilize the learning-based variant. The learned variant operates reliably at that scale and reaches 3.4868, but performs marginally worse than the reported 3.48 tuning-free figure. Accuracy figures for LLaMA-2-7B and LLaMA-3-8B reflect unified harness evaluations from Table 8, while the remaining four columns preserve earlier protocols; as shown in Appendix B, this difference shifts aggregate scores by up to 1 point. Superscript ‡ indicates measurements taken on NVIDIA B200 rather than H200; hardware offsets are quantified in Appendix B, and this entry lacks baselines in the literature.

C.5 Extended Motivation

This subsection preserves the original narrative for the four observations in §2.

Observation 1. Total Rounding Error Norm is Invariant, and Available Control Levers Decompose into Orthogonal Axes. A b-bit uniform quantizer maps continuous inputs onto discrete lattice points with step size Δ, injecting residual noise with an element-wise variance of $\Delta^2/12$. For a fixed lattice spacing, this residual magnitude is an invariant arithmetic property of rounding rather than an algorithmic choice. Consequently, a quantizer possesses exactly two orthogonal degrees of freedom:

1. **Magnitude Control (Step Size Reduction):** Transforming the coordinate basis of the target tensor to compress its dynamic range and minimize Δ (e.g., orthogonal rotations that suppress outlier peaks, or Lloyd-Max companding matched to source distribution density).
2. **Directional Relocation (Subspace Steering):** Steering the fixed residual energy into the attenuating subspace of downstream linear operators.

Crucially, these two levers are mathematically decoupled because they manipulate distinct quantities. Projecting error onto the downstream eigenbasis (Figures 4 and 5) reveals that rotating prior to rounding suppresses *downstream distortion* $\|\mathbf{W}(\mathbf{x} - \hat{\mathbf{x}})\|^2$ by an order of magnitude, yet leaves the *proportion* of error energy carried by the most sensitive top 10% eigenspace strictly at its isotropic baseline. It lowers the entire carpet without altering its shape. Sequential relocation operates in reverse: metric distortion decreases moderately, but the proportion in critical subspaces is suppressed by an order of magnitude. These quantities need not move in the same direction: relocation can halve downstream distortion even while inflating the raw error energy (Figure 4). Detailed measurements are provided in §4.6.

In summary, the former governs error *magnitude* while the latter governs its *geometric orientation*; neither can substitute for the other, and only their composition achieves the joint gains documented in §4.6. Recent rotation-based frameworks (Ashkboos et al., 2024b; Liu et al., 2025; Hu et al., 2025; Sun et al., 2025) have plateaued precisely because they exploit only magnitude suppression without formalizing directional relocation.

Observation 2. Downstream Operators are Anisotropic, and Admit Exact Closed-Form Hessians in Key Primitives. In deep architectures, quantized tensors are never consumed in isolation; they serve as inputs to downstream linear operations $\mathbf{A}$ (e.g., weight multiplications $\mathbf{W}$ for activations, or attention-weighted sums for value caches). Because $\mathbf{A}$ is linear, the end-to-end output distortion is governed strictly by the quadratic form $\mathcal{D} = \mathbb{E}[(\mathbf{x} - \hat{\mathbf{x}})^\top \mathbf{H}(\mathbf{x} - \hat{\mathbf{x}})]$ parameterized by the Hessian $\mathbf{H} = \mathbf{A}^\top \mathbf{A}$. The eigenspectrum of $\mathbf{H}$ determines both the attenuating null spaces where error is safely absorbed and the sensitive principal directions where error must be suppressed.

Standard element-wise MSE minimization is optimal if and only if $\mathbf{H} \propto \mathbf{I}$. Across the core primitives of transformers, however, $\mathbf{H}$ is strictly anisotropic, deviating from the flat spectrum assumed by element-wise MSE by over three orders of magnitude (Appendix Figure 5). Along the activation path in particular, the downstream Hessian $\mathbf{H} = \mathbf{W}^\top \mathbf{W}$ is **analytically determined in closed form** directly from static layer weights, requiring zero calibration data. This contrasts with GPTQ (Frantar et al., 2023), where the weight Hessian $\mathbb{E}[\mathbf{X}\mathbf{X}^\top]$ must be approximated over calibration samples, allowing our activation compensation to be entirely data-free (§3.2).

Observation 3. Existing Paradigms Operate via Implicit, Single-Axis Relocation Heuristics. Reinterpreting prior literature through this geometric lens reveals that prevailing approaches perform

implicit error relocation along ad-hoc, heuristically chosen axes. For example, GPTQ sequentially propagates rounding residuals onto unquantized channels (the channel axis); SmoothQuant (Xiao et al., 2023) and AWQ (Lin et al., 2024) transfer dynamic range difficulty from activations to weights via diagonal equivalence transformations (the scale axis); and rotation-based methods disperse spatial outliers uniformly (the basis transformation axis).

However, prior works failed to formalize quantization as a unified relocation objective, leaving the following foundational question unanswered:

> *"What are the attenuating subspaces of a given downstream operator, and how can the theoretical utility of relocating error along candidate transformation axes be quantified* a priori *prior to quantization?"*

Lacking such a predictive formulation, prior methods were forced to identify transformation axes through empirical trial-and-error, expending computation on ineffective dimensions while forfeiting high-gain structural pathways.

Observation 4. Theoretical Gains Across Candidate Axes are Predictable *A Priori* from Operator Statistics, Dictating Distinct Strategies per Tensor. Even within the same attention block, key (K) and value (V) tensors demand diametrically opposed optimization strategies, as evidenced in Figure 6:

1. **Key (K) Cache:** Outliers in K activations are dominated by stationary per-channel DC offsets (median per-head crest factor 6.31). Simply subtracting a calibration-derived mean vector reduces the crest factor to 3.27. Composing this centering with a per-head orthogonal Hadamard transform disperses remaining peaks uniformly, enabling near-lossless 3-bit key quantization (5.92 vs. 5.73 FP16 PPL).
2. **Value (V) Cache:** Conversely, the majority of V heads are approximately Gaussian (crest factor 2.95), yet the top 1% of extreme heads forms a severe heavy tail with a 99th percentile of 10.19. Coarse metrics that average kurtosis across all heads obscure these localized bottlenecks, necessitating per-head processing.
3. **Temporal Dimension:** Because attention-weighted averaging exhibits intrinsic low-pass filtering properties, Σ-Δ noise shaping could theoretically exploit this structure. However, the theoretical ceiling on error suppression is dictated by the lag-1 autocorrelation ρ of attention weights. In autoregressive decoders, empirical autocorrelation is negligible ($\bar{\rho} \approx 0.05$). The theoretical headroom along the temporal axis is therefore virtually zero, and the optimal strategy is to deliberately bypass temporal operations to avoid superfluous runtime overhead.

In this manner, OptQuant systematically derives distinct operations across tensor paths (namely, **exact closed-form compensation**, **combined mean-centering and rotation**, and **deliberate pruning of ineffective dimensions**), all originating from proactive statistical profiling rather than post-hoc tuning.

C.6 Figures Supporting Motivational Observations

The following three figures provide empirical support for the observations in §2; numerical quantities are detailed in the corresponding main text sections.

C.7 Weight-Only Quantization Across All Five Evaluated Models

Table 7 restricts comparisons to the four models reported by OSAQ to enable direct evaluation within their grid. Table 6 reports our own measured weight-only bit ladder across all five primary evaluation models. Granularities adhere to OSAQ conventions: `W4` is per-channel, while `W3` and `W2` utilize group size 128. Zero-shot columns report 9-task macro-averages evaluated at group size 128 across all bitwidths; the `W4` accuracy column therefore reflects a finer granularity than the corresponding perplexity column in the same table. The task-level breakdown (Table 2) and Table 7 follow OSAQ's grid strictly, having re-evaluated `W4` and `W3` per-channel for exact alignment.

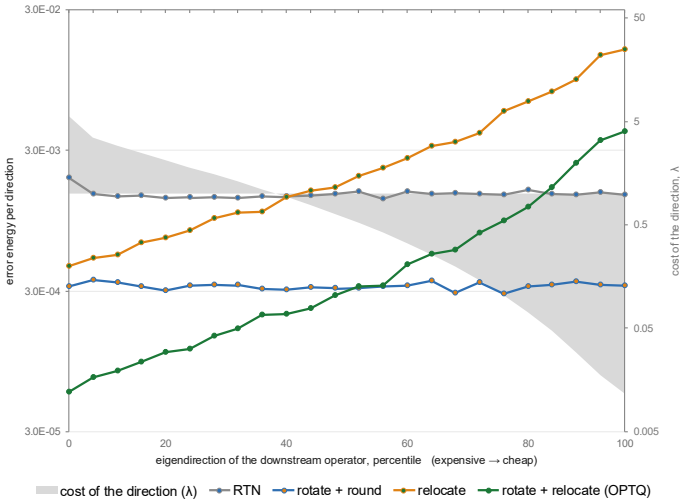


Figure 4: Quantization error is steered rather than eliminated. Evaluated on LLaMA-2-7B layer 16 `o_proj`, 4 bits, across 40 tokens. Eigen-directions are arranged from most expensive to least expensive; the gray band denotes the cost λ_i (right axis). While RTN and rotations leave error placed identically to white noise, relocation mirrors the cost curve, halving output distortion while bearing $2.53\times$ the raw error energy of RTN on this module.

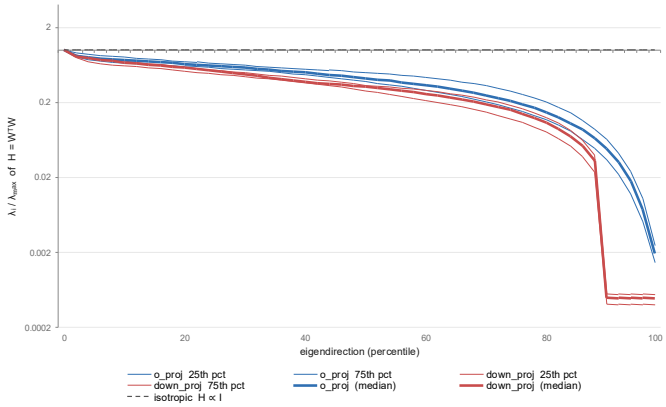


Figure 5: Downstream operators are strictly anisotropic. Eigenspectrum of $\mathbf{H} = \mathbf{W}^\top\mathbf{W}$ (median and interquartile range) evaluated across LLaMA-2-7B. The flat spectrum implicitly assumed by element-wise MSE (dashed line) deviates from reality by over three orders of magnitude; this anisotropy represents the exact headroom available for error relocation.

This ladder reproduces the trends documented in Appendix C.11 along the weight-only axis: 4-bit compression is virtually lossless (within 0.9 points of FP16 accuracy across all models); 3-bit incurs modest degradation; and 2-bit separates models by parameter capacity rather than collapsing them uniformly. This contrasts with joint W-A-KV quantization, where 2-bit accuracy converged onto a narrow band regardless of model scale.

C.8 Compensation Factor Precision

Relocation maintains a static $D \times D$ compensation factor in memory per quantized module throughout execution, making factor precision both a numerical stability concern and a memory scalability concern. Halving precision from FP32 to FP16 shifts `W4A4KV4` perplexity on LLaMA-2-7B from 5.714 ± 0.007 (three-seed FP32) to 5.7148 (FP16). This represents a minor 0.09σ shift that falls comfortably within seed variance ($[5.7091, 5.7224]$). Consequently, storing factors in half-precision incurs no measurable accuracy degradation. This is operationally significant because FP32 footprint

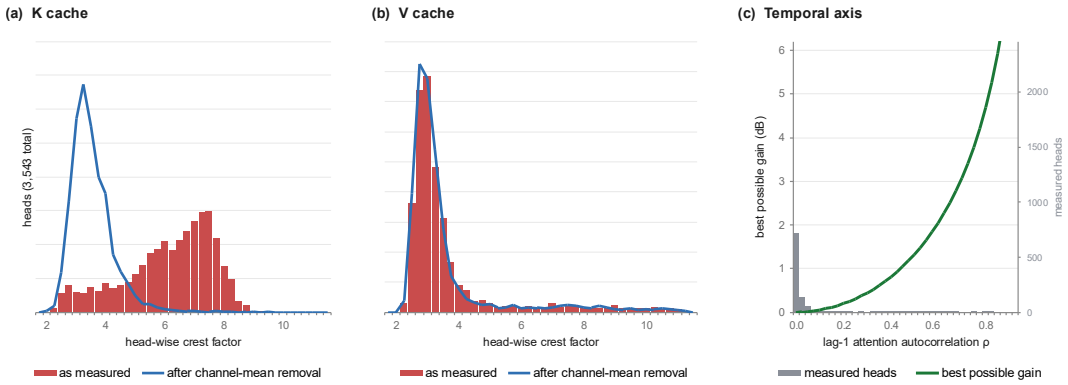


Figure 6: Why distinct axes demand distinct treatments. **Left and Center:** Crest factor distributions across 3,543 heads in LLaMA-2-7B. Channel-mean centering reduces heads with crest factor > 6 from 57% to 1% in key caches, but shifts value caches marginally from 9% to 8%, where the extreme tail of V ($p_{99} = 10.2$) remains masked by aggregate metrics. **Right:** Temporal axis. Proposition 3 bounds achievable gain by $-10\log_{10}(1-\rho^2)$. Half of all heads fall below $\rho = 0.01$, and achieving a modest 0.5 dB gain requires $\rho \geq 0.33$, a threshold satisfied by only 5.7% of heads.

Table 6: Directly evaluated weight-only quantization across all primary models. Perplexity is evaluated on the complete WikiText-2 test split and C4 under the standard GPTQ protocol; accuracy represents 9-task zero-shot macro-averages.

	WikiText-2 PPL ↓				C4 PPL ↓				9-task Acc (%) ↑			
Model	FP16	W4	W3	W2	FP16	W4	W3	W2	FP16	W4	W3	W2
LLaMA-3.2-1B	9.755	10.196	11.719	26.817	14.052	15.281	18.991	64.213	55.75	54.89	51.72	39.03
LLaMA-3.2-3B	7.815	8.044	8.856	16.366	11.367	12.064	14.148	34.407	63.44	62.82	59.90	46.85
LLaMA-2-7B	5.468	5.607	5.849	8.510	7.265	7.525	8.173	14.057	65.08	64.61	62.48	53.67
LLaMA-3-8B	6.139	6.400	7.101	12.602	9.475	10.118	11.988	28.145	68.19	68.00	65.85	51.45
LLaMA-2-13B	4.880	4.948	5.150	6.622	6.729	6.860	7.327	10.716	67.81	67.70	66.44	60.37

reaches 365 GB on 70B models (§B.3). All primary results reported in the main text utilize FP32 factors. At 70B the same comparison was measured directly on the `W3A3KV3` cell: halving factor precision shifted the 9-task average by 0.13 points (69.32 FP32 vs. 69.19 FP16) and individual tasks by up to 1.02, matching the magnitude of the batch-size effect (Table 5).

C.9 Baseline Protocol Notes

Protocol Notes on Self-Reported Baselines. FlatQuant's headline result (LLaMA-3-8B `W4A4` = 6.90) omits KV cache quantization; its actual `W4A4KV4` perplexity reported in its appendix is 7.09. Even when compared against self-reported figures, our method (6.90) is strictly superior, and on 7B our `KV4` perplexity (5.72) surpasses their unquantized `KV16` result (5.78).

On C4, our raw perplexities trail the OSAQ family under `W4` and `W3`. However, this comparison breaks down upon inspecting baseline calibration: Table 9 places unquantized *FP16* reference numbers side-by-side. While aligning within 0.036 across all four models on WikiText-2, our FP16 baselines on C4 are higher by 0.27 to 0.29 on the three smaller models and by 0.19 on 70B. Because no quantization is executed under FP16, this penalty is imposed on our evaluation prior to rounding a single weight tensor, remaining virtually constant across models. This represents a protocol discrepancy rather than an algorithmic property.

We investigated the most plausible hypothesis: OSAQ does not specify evaluation sequence length, and LLaMA-2 models support a 4096 context window where longer evaluation windows naturally yield lower perplexity. Repartitioning the identical token stream to sequence length 4096 reduces C4 perplexity from 7.263 to 7.099, which remains 0.13 higher than their 6.97, while breaking the WikiText-2 alignment that justified comparison in the first place (5.114 vs. their 5.47, which our 2048 configuration matched at 5.467). Their evaluation sequence length is therefore 2048 as well,

Table 7: Complete weight-only perplexity grid under OSAQ configurations, evaluating WikiText-2 and C4 per model. The main text reports weight-only performance in terms of accuracy (Table 2); this table provides corresponding perplexities. † denotes results cited from OSAQ, and `-cd` indicates their coordinate-descent variant. Empty cells indicate un-reported configurations in the original work. Our rows are highlighted with shading. C4 columns cannot be compared directly between studies (Table 9); Appendix C.9 interprets them via degradation relative to respective FP16 baselines.

		LLaMA-2-7B		LLaMA-2-13B		LLaMA-3-8B		LLaMA-2-70B	
Config	**Method**	Wiki	C4	Wiki	C4	Wiki	C4	Wiki	C4
	FP16 (ours)	5.47	7.27	4.88	6.73	6.14	9.47	3.32	5.71
W4A16	RTN†	6.11	7.71	5.20	6.83	8.70	14.00	3.67	5.79
	MagR†	5.91	7.52	5.17	6.81	–	–	3.58	5.72
	OmniQuant†	5.74	7.35	5.02	6.65	–	–	3.47	5.65
	QuIP†	5.94	8.01	5.01	6.88	6.50	11.10	3.53	5.87
	AWQ†	6.15	7.68	5.12	6.74	7.10	10.10	3.57	5.71
	OSAQ+AWQ†	5.99	7.50	5.04	6.67	6.82	9.93	3.53	5.66
	GPTQ†	5.83	7.37	5.13	6.70	7.00	11.80	3.58	5.67
	OSAQ+GPTQ†	5.73	7.34	5.04	6.64	6.82	11.50	3.48	5.61
	OptQuant	**5.61**	**7.53**	**4.95**	**6.86**	**6.40**	**10.12**	**3.39**	**5.77**
W3A16-g128	RTN†	6.66	8.40	5.51	7.18	27.90	110.00	3.97	6.02
	MagR†	6.46	8.22	5.45	7.12	–	–	3.95	6.00
	OmniQuant†	6.03	7.75	5.28	6.98	–	–	3.78	5.85
	AWQ†	6.24	7.84	5.32	6.94	8.20	11.60	3.83	5.82
	OSAQ+AWQ†	6.08	7.75	5.23	6.89	7.96	11.40	3.75	5.72
	GPTQ†	6.29	7.89	5.42	7.00	8.20	13.70	3.85	5.85
	OSAQ+GPTQ†	6.07	7.77	5.30	6.95	7.98	13.40	3.79	5.78
	OptQuant	**5.85**	**8.17**	**5.15**	**7.33**	**7.10**	**11.99**	**3.61**	**6.01**
W3A16	RTN†	539.00	402.00	10.70	12.50	2200	560.00	7.52	10.00
	MagR†	8.66	10.8	6.55	8.26	–	–	4.64	6.77
	OmniQuant†	6.58	8.65	5.58	7.44	–	–	3.92	6.06
	AWQ†	24.00	23.90	10.50	13.10	12.80	16.80	6.29	7.11
	GPTQ†	8.37	9.81	6.44	8.02	13.00	45.90	4.82	6.57
	OSAQ+GPTQ†	6.75	8.70	5.72	7.54	11.30	23.10	4.21	6.09
	QuIP†	6.50	8.74	5.34	7.34	7.50	11.30	3.85	6.14
	OSAQ+QuIP†	6.37	8.68	5.25	7.27	7.43	10.80	3.81	6.06
	OptQuant	**6.44**	**8.97**	**5.16**	**7.33**	**7.17**	**12.11**	**3.62**	**6.03**
W2A16-g128	RTN†	12000	72000	5800	3900	1900	25000	7800	3600
	MagR-cd†	9.94	14.10	7.63	10.60	–	–	5.52	8.05
	OmniQuant†	11.10	15.00	8.26	11.10	–	–	6.55	8.52
	GPTQ†	36.80	33.70	28.10	21.00	210.00	41000	19.20	13.80
	OSAQ+GPTQ†	21.20	18.30	13.40	13.80	63.80	352.00	10.70	9.44
	OSAQ+GPTQ-cd†	10.60	14.70	7.60	10.50	26.10	62.00	5.97	8.42
	OptQuant	**8.51**	**14.06**	**6.62**	**10.72**	**12.60**	**28.14**	**4.74**	**7.86**

and the remaining discrepancy stems from *which* C4 documents were scored; the original paper does not specify shard indices or document counts with reproducible precision.

When unquantized baselines diverge, meaningful comparison is possible only via degradation relative to respective FP16 baselines. Under this reading, C4 results flip in our favor at 4 bits: on 2-7B, 2-13B, 3-8B, and 2-70B, our degradations are $+0.260$, $+0.131$, $+0.643$, and $+0.066$, compared to $+0.370$, $+0.180$, $+0.730$, and $+0.090$ for the strongest OSAQ baseline. At 2 bits, our advantage expands substantially ($+18.67$ vs. $+52.80$ on 3-8B; $+2.16$ vs. $+2.90$ on 70B). A deficit persists only at `W3A16`-g128 across all four models ($+0.908$, $+0.599$, $+2.514$, and $+0.300$ vs. $+0.780$, $+0.430$, $+2.200$, and $+0.200$). The fact that this single deficit appears consistently from 7B to 70B demonstrates that it is structural rather than random noise. We attribute this to unaligned configuration choices: OSAQ quantizes weights asymmetrically with explicit zero-points whereas our scheme is symmetric, and asymmetric grids offer superior expressive capacity in intermediate bit regimes where dynamic range utilization is critical.

Calibration corpus shifts represent a secondary factor. Re-calibrating across all six affected cells using C4 instead of WikiText-2 recovers 4% of the raw deficit on 2-13B at `W4` and 79% on 3-

8B at W3 (averaging 26%). While measurable, this is a minor effect that cannot explain baseline discrepancies already present in FP16.

C.10 Pipeline Architecture and System Integration

The optimal operators are integrated into a unified inference pipeline:

- **Weights:** Block-Hadamard incoherent transformation $\rightarrow \mathcal{Q}_{\text{reloc}}$ sequential error compensation $\rightarrow$ Lloyd-Max Gaussian companding (§3.3).
- **Activations:** Sensitive pathways (o_proj, down_proj; §3.6) receive ROTA-STE transformations; remaining linear layers combine static Hadamard rotations, Fast Mode weight-matched relocation (§3.2), and Gaussian companding. (Companding assumes a Gaussian source and is deferred on the two ROTA-STE projections, as their activations violate this assumption prior to learning transformations; §4.5.)
- **Key (K) Cache:** Composed mean centering and per-head Hadamard transforms (§3.4).
- **Value (V) Cache:** Standard per-head RTN (temporal feedback pruned via Proposition 3; §3.5).

All compensation matrices, rotation operators, and quantization lookup tables are precomputed offline. Dynamic serving overhead is strictly confined to block GEMMs in Fast Mode relocation and fast Walsh-Hadamard transforms (runtime benchmarks in §C.13).

C.11 Bitwidth Ladder

Table 10 illustrates that perplexity degradation is graceful from 4 to 3 bits, but collapses sharply at 2 bits. In particular, 2-bit zero-shot accuracy converges onto an invariant band between 30.8% and 31.9% across a 13× model scale range (1B to 13B), indicating that 2-bit weights represent an absolute information bottleneck under linear representations.

C.12 Architectural Diversity and Scale Generalization

We apply our tuning-free configuration across four distinct model families spanning 1B to 70B parameters. Because tokenizers differ across families, absolute perplexities cannot be compared directly across architectures; evaluations focus on relative degradation relative to unquantized FP16 baselines.

Supplementary Evaluation on EXAONE Architectures. Under weight-only W3/W2 compression, EXAONE-4.0-1.2B degrades by +10.7% (W3) and +65.5% (W2) on WikiText-2, while EXAONE-4.5-33B degrades by +7.1% (W3) and +41% (W2) on WikiText-2 (+7.4% / +58.5% on C4). This confirms that scale-dependent resilience patterns documented in Appendix C.11 reproduce on a third independent architecture. Zero-shot evaluation for EXAONE applies its native chat template and reports the 8-task average excluding LAMBADA; empirical motivations for both choices are detailed in Appendix C.14. Under this protocol, the model retains 96.4% of FP16 accuracy at W4A4KV4.

For EXAONE-4.5-33B (a vision-language checkpoint), quantization is applied strictly to the language backbone while the vision tower remains at native precision. Large models (33B and 70B) utilize tuning-free pipelines; as noted in Appendix B.3, this is not because learned variants fail at scale, but because learning does not justify its computational cost. Finally, the Gemma-2 family was excluded due to an interaction between logit soft-capping and evaluation harnesses that precluded fair FP16 calibration.

C.13 Serving Efficiency: Runtime Profiling and Memory Footprint

While developing dedicated INT4 GEMM kernels lies outside our scope, we profile runtime overhead introduced by relocation operators on NVIDIA H200 GPUs.

As shown in Table 13a, compiled execution via `torch.compile` reduces total relocation overhead to 16.5% of layer GEMM execution time at sequence length 32k. Table 13b reports a negative engineering outcome: three hand-optimized fused Triton kernels did not outperform compiler-generated code (standalone bitwise-exact companding kernels achieved $11\times$ to $23\times$ speedups, translating to an end-to-end reduction of 3.1%).

Memory allocations are reported analytically in Table 13c. Comparing under iso-memory conditions reveals that allocating bit budgets to smaller models at higher precisions is Pareto-optimal: a 3B model at `W4A8KV8` achieves 62.67% accuracy, whereas a 7B model at `W3A3KV16` reaches only 60.05% (Table 10). However, because evaluation context lengths differ substantially across these two setups, this comparison holds across aggregate memory footprint rather than matched sequence lengths.

C.14 Lessons from Evaluation Methodology

First, zero-shot protocols for instruction-tuned models depend heavily on task *type*, and neither raw text nor templated evaluation is universally safe as a default. Evaluating EXAONE-4.0-1.2B under both protocols reveals that without chat templates, five of eight multiple-choice tasks languish at chance performance even under FP16 (deviations between -0.1 and $+0.4$ points). Applying chat templates elevates all eight tasks by 3.8 to 28.2 points above chance. However, templates structurally invalidate tasks requiring exact string continuation such as LAMBADA, because assistant turn termination tokens halt generation immediately (0.0%). No single protocol satisfies both task families simultaneously; we adopt templates while omitting LAMBADA, reporting both sets of measurements in the appendix. The fact that raw WikiText-2 perplexity on this model reaches 55 (approximately $5.6\times$ higher than contemporary 1B base models) reinforces this finding: when evaluation formats misalign with training objectives, evaluation collapses before quantization is even applied. Second, Needle-In-A-Haystack (NIAH) saturates completely under 4-bit KV compression, with all three evaluated methods scoring exactly 0.960. Omitting invariant metrics encourages selective benchmarking; we therefore report this finding explicitly as a null result.

C.15 Supplementary Tables

The following tables support the analyses in §4, providing task-level zero-shot breakdowns, batch size and hardware controls, K×V precision grids, architectural ladders, runtime and memory profiles, and operator ablations. Each table is discussed where cited in the main text or appendix.

Table 8: Task-level zero-shot accuracy under `W4A4KV4` evaluated over full test splits, grouping each model into a distinct row block. All rows were measured on our hardware using a unified harness (`lm-evaluation-harness` 0.4.12); baselines represent our direct reproductions using official author code rather than cited figures, ensuring direct column-wise comparability. Perplexities are reported under identical protocols where evaluated. Evaluation batch size is not uniform across rows, as each baseline retains its native batch size (FP16 and OptQuant 64, RTN 8, QuaRot 16, SpinQuant 4). Batch size is not entirely benign: holding models, harnesses, and seeds fixed, shifting batch size from 32 to 64 alters SIQA by 0.36 points on LLaMA-2-7B and 0.57 points on LLaMA-3-8B. Appendix B measures this across all nine tasks, showing that this variance is an order of magnitude smaller than the algorithmic margins evaluated here.

		PPL	Per-Task Accuracy (%) ↑									Avg_9
Model	**Method**	↓	BoolQ	PIQA	SIQA	HellaS.	WinoG.	ARC-e	ARC-c	OBQA	LAMB.	↑
LLaMA-2 7B	FP16	5.467	79.39	78.29	46.62	76.20	69.30	73.91	44.62	44.0	73.45	65.09
	Naive RTN	17.702	40.70	50.87	34.95	26.45	49.57	25.93	26.19	24.4	0.02	31.01
	QuaRot	6.367	74.68	76.61	44.32	71.92	65.82	68.43	42.32	41.0	67.71	61.42
	SpinQuant	6.600	71.59	75.46	44.58	69.42	63.30	67.34	40.19	38.8	66.52	59.69
	OptQuant	**5.722**	**76.39**	**78.29**	**46.52**	**74.49**	**67.80**	**70.79**	**43.60**	**43.8**	**72.19**	**63.76**
LLaMA-3 8B	FP16	6.136	82.17	80.41	46.52	79.29	73.48	77.57	54.01	44.8	75.55	68.20
	Naive RTN	141.54	50.24	54.84	34.14	30.20	52.25	29.17	24.06	26.8	4.58	34.03
	QuaRot	8.877	70.76	74.76	43.60	71.32	66.46	67.85	42.83	39.4	64.22	60.13
	SpinQuant	8.150	73.24	76.17	43.71	74.36	67.96	73.44	45.73	39.8	68.62	62.56
	OptQuant	**6.899**	**81.01**	**78.40**	**46.62**	**77.35**	**72.61**	**77.48**	**52.05**	**44.8**	**73.47**	**67.09**

Table 9: Cross-corpus C4 comparisons rest on misaligned unquantized baselines. We list unquantized FP16 baselines from both studies alongside sequence-length probe experiments conducted on LLaMA-2-7B over the identical token stream. While WikiText-2 aligns closely, C4 diverges by a virtually constant offset. Repartitioning to length 4096 does not reconcile C4 and disrupts WikiText-2 alignment, ruling out sequence length as the source of discrepancy. Our entries in this table originate from a separate measurement pass than Table 6, accounting for minor third-decimal differences (5.4668 vs. 5.468); each table is internally consistent and comparisons must be read within tables.

	WikiText-2 FP16			C4 FP16		
Model	OSAQ	Ours	Δ	OSAQ	Ours	Δ
LLaMA-2-7B	5.47	5.4668	-0.003	6.97	7.2631	$+0.293$
LLaMA-2-13B	4.88	4.8789	-0.001	6.46	6.7266	$+0.267$
LLaMA-3-8B	6.10	6.1362	$+0.036$	9.20	9.4709	$+0.271$
LLaMA-2-70B	3.31	3.3180	$+0.008$	5.52	5.7088	$+0.189$
Sequence-length probe, LLaMA-2-7B, same token stream						
`seqlen` 2048	5.4668 (OSAQ 5.47)			7.263 (OSAQ 6.97)		
`seqlen` 4096	5.1137 (OSAQ 5.47)			7.099 (OSAQ 6.97)		

Table 10: Bitwidth ladder following the format of Table 1, with model architectures grouped in columns and bit configurations arranged in rows. All entries reflect our own direct measurements. FP16 rows correspond to the cited reference baselines (†) used in Table 1, which our direct evaluations match within ± 0.01 perplexity. Performance degrades gracefully from 4 to 3 bits but collapses at 2 bits, where zero-shot accuracy converges onto a narrow band irrespective of parameter scale.

	LLaMA-2 7B		LLaMA-2 13B		LLaMA-3 8B		LLaMA-3.2 1B		LLaMA-3.2 3B	
Bits	PPL ↓	Acc ↑	PPL ↓	Acc ↑	PPL ↓	Acc ↑	PPL ↓	Acc ↑	PPL ↓	Acc ↑
`16-16-16`†	5.47	65.08	4.88	67.77	6.13	68.05	9.76	55.75	7.81	63.44
`W4A8KV8`	5.57	64.44	4.95	66.91	6.42	67.83	10.27	54.81	8.06	62.67
`W4A4KV16`	5.64	63.87	5.01	66.67	6.63	67.08	10.78	53.31	8.32	61.90
`W3A3KV16`	6.40	60.05	5.55	64.67	8.71	59.99	17.12	45.04	11.16	54.93
`W2A2KV16`	180.7	31.52	62.96	31.87	1032.8	30.80	5205	31.34	2423	31.30

Table 11: Asymmetric K × V cache grid on LLaMA-2-7B under fixed `W4A4` weights and activations (WikiText-2 PPL ↓).

PPL	V=4	V=3	V=2	V=2, centered + rotated
K=4	5.700	5.819	11.420	**7.162** (−37%)
K=3	5.841	5.967	12.085	**7.497** (−38%)
K=2	6.564	6.813	18.910	**10.745** (−43%)

Table 12: Architectural and scale diversity evaluated via WikiText-2 perplexity. Values must be interpreted as relative degradation against each model's unquantized FP16 baseline, because differing tokenizers preclude cross-family comparison of absolute numbers. The largest two models were evaluated under multi-accelerator sharding, which is mandatory for the 70B checkpoint. The **Configuration** column indicates the operational variant deployed per row. The learning-based variant was evaluated on 70B but performed slightly worse (3.4868), which is why it is not reported as the primary configuration.

Model Family	FP16	W4A16	W4A4KV4	Configuration
LLaMA-3.2-1B	9.76	10.25 (+5.1%)	11.73 (+20.3%)	Learned
LLaMA-2-7B	5.47	5.57 (+1.9%)	5.72 (+4.7%)	Learned
LLaMA-2-70B	3.318	3.386 (+2.0%)	3.478 (+4.8%)	Training-free
Qwen2.5-7B	6.851	6.977 (+1.8%)	7.503 (+9.5%)	Training-free
Qwen2.5-14B	5.292	5.538 (+4.6%)	6.168 (+16.6%)	Training-free
Qwen3-8B	9.718	9.857 (+1.4%)	10.547 (+8.5%)	Training-free
EXAONE-4.0-1.2B	55.28	55.89 (+1.1%)	61.39 (+11.1%)	Learned
EXAONE-4.5-33B	8.265	8.403 (+1.7%)	8.547 (+3.4%)	Training-free

Table 13: Runtime efficiency and memory footprint. (a) Activation epilogue runtime overhead relative to baseline layer GEMM execution on NVIDIA H200. (b) Custom-written fused Triton kernels at sequence length 32k, reported as a negative engineering outcome relative to `torch.compile`. (c) Analytically derived memory requirements.

(a) Epilogue Overhead (Ratio relative to GEMM)

Implementation	4k	16k	32k
Hadamard only	0.138	0.097	0.087
Per-token quantization	0.981	0.866	0.724
Combined, eager	1.082	0.860	0.793
Combined, compiled	**0.261**	**0.182**	**0.165**

(b) Fused Kernels at 32k (Relative to compiled)

Kernel	Ratio	Speedup
`torch.compile` (ref.)	0.165	1.00×
Fused v1	0.214	0.77×
Fused v2 (single pass)	0.228	0.72×
Fused v2 + 4-bit packing	0.559	0.29×

(c) Analytical memory requirements (not empirical allocations). The final two columns indicate resident compensation factors required by relocation that rotation-based frameworks do not incur. These represent algorithmic overheads that scale with model dimension rather than checkpoint size. Extrapolating to LLaMA-2-70B yields 365 GB in FP32 and 182 GB in FP16, substantially exceeding the ∼ 35 GB footprint of `W4` weights.

					Compensation Factors	
Model	**FP16 Weights**	**W4**	**W3**	**KV4 / 1k Tokens**	**FP32**	**FP16**
LLaMA-3.2-1B	1.95 GB	0.50 GB	0.38 GB	8.4 MB	5.5 GB	2.8 GB
LLaMA-3.2-3B	5.64 GB	1.45 GB	1.10 GB	29.6 MB	12.9 GB	6.5 GB
LLaMA-2-7B	12.95 GB	3.34 GB	2.53 GB	135.2 MB	26.5 GB	13.2 GB
LLaMA-3-8B	13.96 GB	3.60 GB	2.73 GB	33.8 MB	36.5 GB	18.2 GB
LLaMA-2-13B	25.38 GB	6.54 GB	4.96 GB	211.2 MB	51.9 GB	26.0 GB

Table 14: Ablation studies on LLaMA-2-7B (unquantized FP16 PPL $= 5.467$). (a) Step-wise addition of operators across the ladder. Weight-only columns in the final three rows are empty because those operators operate strictly along the activation path and have no weight-only counterpart; these entries are undefined rather than missing. (b) Controlled ablations on weight-only `W4`.

(a) Operator Ladder

#	Added Operator	W4A4KV4	Δ	Weight-Only
L1	GPTQ weights + Hadamard activations	6.050	–	5.783
L2	+ weight incoherence (block-Hadamard)	5.912	−0.138	5.628
L3	+ weight Gaussian companding (Lloyd-Max)	5.848	−0.064	5.569
L5	+ ROTA-STE learned activations (`o`/`down`)	5.780	−0.068	–
L6	+ activation/KV Gaussian companding	**5.722**	−0.058	–
L4	*Negative control*: companding without rotation	6.180	+0.332	–

(b) Sensitivity Analysis. The first row reports three-seed `W4A4KV4` results; remaining rows evaluate weight-only `W4`.

Control Variable	PPL		
Final configuration, 3 seeds	5.714 ± 0.007		
	none	*random orth.*	*Hadamard*
Transformation	5.9155	5.6071	**5.5904**
	64	*128*	*256*
Weight group size	5.5612	**5.5596**	5.5657
	WikiText-2	*C4*	
Calibration corpus	**5.5596**	5.5949	
	32 chunks	*128 chunks*	
Calibration budget, unrotated	**5.7833**	5.9155	
Calibration budget, rotated	5.6278	**5.5904**	