

KernelQuant: Post-Training Quantization via Representation Kernel Preservation

Juneyoung Park, Youngwook Kwon, Seongwan Kim[†], Jaeho Lee[†]

OptAI Inc.

{jyoung.park, ywook.kwon, swan.kim, jaeho.lee}@opt-ai.kr

Abstract

Post-training quantization methods universally minimize element-wise reconstruction error, yet we show this is the wrong objective. A systematic correlation study on LLaMA models reveals that the *representation kernel* (the matrix of pairwise cosine similarities among calibration inputs) is a substantially stronger proxy for downstream task performance than weight MSE or output MSE. Building on this finding, we propose KernelQuant, a training-free PTQ framework comprising: KPQ, which selects weight-quantization scales via kernel-MSE grid search followed by least-squares error compensation; model-level Hadamard rotation; and KPC, a kernel-preserving per-channel activation clipping scheme. KernelQuant introduces zero inference overhead beyond QuaRot and calibrates in minutes rather than hours. On LLaMA 3B–70B, KernelQuant achieves state-of-the-art W4A16 perplexity and competitive W4A4 results. On reasoning benchmarks (MMLU), where 4-bit quantization causes a “reasoning cliff,” KernelQuant retains the highest fraction of FP16 accuracy among all evaluated baselines, both rotation-based (QuaRot, SpinQuant, FlatQuant) and element-wise, reversing the perplexity-based ranking and supporting the view that reasoning depends on the geometric relations the kernel preserves.

1 Introduction

Post-training quantization (PTQ) has emerged as the dominant approach to compress large language models (LLMs) without retraining, with RTN, GPTQ (Frantar et al., 2023), and AWQ (Lin et al., 2024) becoming de facto standards for weight-only compression, and rotation-based methods such as QuaRot (Ashkboos et al., 2024), SpinQuant (Liu et al., 2025b), and FlatQuant (Sun et al., 2024) achieving state-of-the-art results when activations are quantized jointly with weights.

Despite their diversity, these methods share a single implicit assumption: *the goal of quantization is to minimize element-wise error*, whether measured as weight MSE (RTN), output MSE (GPTQ), activation-weighted MSE (AWQ), or reconstruction error under learned transformations (FlatQuant, SpinQuant). In this work we revisit this assumption. We conduct a systematic correlation study on four LLaMA family models (Section 5.1; Appendix F) and find that, whenever the proxy metrics diverge, the degree to which the *representation kernel* (the matrix of pairwise cosine similarities among calibration inputs) is preserved correlates more strongly with downstream task performance than either weight MSE or output MSE. Even when element-wise errors are small, kernel degradation leads to substantial performance drops; conversely, larger element-wise errors are tolerable as long as the kernel is preserved (Figure 1).

We argue that quantization should preserve the relational structure the model has learned through training, not the precision of individual weights, consistent with recent observations on representation convergence across networks (Huh et al., 2024). We therefore propose KernelQuant, a PTQ framework that redefines the quantization objective in terms of kernel preservation. KernelQuant consists of (i) KPQ, combining kernel-preserving scale search with least-squares error compensation; (ii) model-level Hadamard rotation absorbed into adjacent weights; and (iii) KPC (Kernel-Preserving Channel scaling), which protects channels of high structural importance. The pipeline is training-free and incurs no inference overhead beyond QuaRot.

KPQ matches or improves over GPTQ on W4A16 perplexity at every scale from 3B to 70B (Table 1), and KernelQuant achieves W4A4 perplexity that is strictly better than SpinQuant and competitive with FlatQuant while avoiding its per-layer affine cost: it surpasses FlatQuant on LLaMA-3-8B and trails by at most 0.13 PPL on LLaMA-2

[†]Corresponding Author

models (Table 2). Calibration completes in minutes via grid search over $C = 80$ candidates, compared to hours for SpinQuant and FlatQuant. KernelQuant achieves the highest MMLU retention of any evaluated W4A4 method, surpassing FlatQuant by 4.75 retention points on LLaMA-3-8B and 3.19 on LLaMA-2-13B (Table 3), and the same algorithm, without modification, yields consistent benefits on vision and audio models (Appendix E).

Contributions. (1) We propose representation-kernel MSE as the PTQ objective, supported by a correlation study showing it tracks downstream performance more closely than weight or output MSE. (2) We introduce KernelQuant: KPQ pairs a kernel-MSE grid search over per-channel base scales (single global shrinkage, C evaluations) with Hessian-based error compensation, yielding per-channel quantization scales that outperform both GPTQ and AWQ without independent per-channel optimization. Combined with Hadamard rotation and KPC activation clipping, the full pipeline is training-free, calibrates in under 10 minutes, and adds zero inference overhead over QuaRot. (3) We evaluate on LLaMA 3B–70B under W4A16 and W4A4, and validate the same algorithm on vision and audio without modification. To delineate our contribution precisely: the kernel-MSE scale search (KPQ Step 1) and the kernel-preserving activation clipping (KPC) are new to this work, whereas the Hessian-based least-squares error compensation (KPQ Step 2) is adopted from GPTQ (Frantar et al., 2023) and the Hadamard/Cayley rotation machinery from QuaRot (Ashkboos et al., 2024) and SpinQuant (Liu et al., 2025b).

2 Related Work

2.1 Weight Quantization

Post-training quantization (PTQ) for weight compression has been extensively studied as a practical route to reducing the memory footprint of LLMs without retraining. Round-to-Nearest (RTN) determines scales from maximum absolute weight values; while negligible in cost, it degrades sharply below 4 bits. GPTQ (Frantar et al., 2023) addresses this via a second-order correction based on the layer-input Hessian, sequentially quantizing weight columns and redistributing reconstruction error to unquantized columns. AWQ (Lin et al., 2024) protects salient channels (identified by large activation magnitudes) through per-channel scaling prior to quantization. OmniQuant (Shao et al.,

2024) jointly learns clipping boundaries and equivalent transformations via block-wise optimization, and QuIP# (Tseng et al., 2024) leverages random orthogonal transforms with lattice codebooks for extreme bit-widths.

All these methods formulate the quantization objective as element-wise reconstruction error. KPQ departs from this convention by optimizing the pairwise cosine similarity matrix (the representation kernel), which captures relational structure across samples and, as we show in Section 5 and Appendix F, is a stronger proxy for downstream task performance whenever it diverges from the element-wise metrics.

2.2 Activation Quantization

Quantizing activations alongside weights enables integer-only arithmetic but poses distinct challenges due to outlier channels. SmoothQuant (Xiao et al., 2023) migrates quantization difficulty from activations to weights via per-channel scaling. A more principled line uses orthogonal transformations: QuaRot (Ashkboos et al., 2024) inserts randomized Hadamard rotations at invariant points in the Transformer, spreading outlier energy uniformly and enabling 4-bit inference; SpinQuant (Liu et al., 2025b) replaces the fixed Hadamard with learned orthogonal matrices on the Stiefel manifold; and FlatQuant (Sun et al., 2024) further generalizes to per-layer learnable affine maps via Kronecker decomposition, achieving state-of-the-art W4A4 performance.

These methods optimize for outlier removal (QuaRot), distribution flatness (FlatQuant), or end-to-end task loss (SpinQuant), without an explicit criterion for which *structural* property of representations should survive quantization. KernelQuant builds on the model-level rotation framework of QuaRot and adds a kernel-preserving objective, targeting a complementary dimension of the problem.

3 Method

3.1 Preliminaries

Throughout the paper, KPQ denotes the full weight method (Step 1: kernel-preserving scale search, Step 2: least-squares error compensation) and KernelQuant is the end-to-end pipeline (KPQ + Hadamard rotation + KPC). In the analysis (Section 5), we additionally isolate Step 1 alone (referred to as KPQ-S, scale-only) as an ablation base-

line to characterize the independent contribution of the kernel-preserving scale search.

Representation Kernel. Consider an arbitrary intermediate layer of a neural network that produces an output matrix $\mathbf{Y} \in \mathbb{R}^{N \times D}$ for N calibration samples. We capture the geometric structure of this representation set via the *cosine similarity kernel matrix*

$$K(\mathbf{Y})_{ij} = \frac{\langle \mathbf{y}_i, \mathbf{y}_j \rangle}{\|\mathbf{y}_i\| \cdot \|\mathbf{y}_j\|} = \cos(\theta_{ij}), \quad (1)$$

where $\mathbf{y}_i$ denotes the i -th row (sample) of $\mathbf{Y}$. The resulting matrix $K(\mathbf{Y}) \in \mathbb{R}^{N \times N}$ encodes all pairwise angular relations among the N samples and is invariant to the magnitude of individual representations. In matrix form it can be written as $K(\mathbf{Y}) = \bar{\mathbf{Y}}\bar{\mathbf{Y}}^\top$, where $\bar{\mathbf{Y}}$ denotes the row-wise ℓ_2 -normalized version of $\mathbf{Y}$.

To quantify the change in kernel structure induced by quantization, we define the *Kernel MSE* between the original full-precision representations $\mathbf{Y}_O$ and the quantized representations $\mathbf{Y}_Q$ as

$$\mathcal{L}_{\text{kernel}}(\mathbf{Y}_O, \mathbf{Y}_Q) = \frac{1}{N^2} \|\mathbf{K}(\mathbf{Y}_O) - \mathbf{K}(\mathbf{Y}_Q)\|_F^2 \quad (2)$$

A smaller value indicates that the pairwise relational structure among samples is better preserved under quantization.

Choice of similarity measure. The cosine kernel is one of several representation-similarity measures; alternatives include CKA and other centered Gram-based statistics (Kornblith et al., 2019). We adopt the cosine kernel for three reasons: (i) it is invariant to per-sample magnitude, matching the error-propagation property that normalization layers cancel magnitude error and propagate only directional error (Appendix D); (ii) it decomposes per-layer and per-channel-group, so it can be evaluated inside a per-layer grid search at $O(N^2D)$ cost without cross-layer statistics; and (iii) unlike CKA, it requires no centering statistics that would themselves be perturbed by quantization. A systematic empirical comparison against alternative structural similarity measures is left to future work.

Uniform Symmetric Quantization. A b -bit uniform symmetric quantizer with scale s is

$$Q_s(x) = \text{clamp}\left(\text{round}\left(\frac{x}{s}\right), -2^{b-1}, 2^{b-1}-1\right) \times s. \quad (3)$$

In RTN, the scale per output channel is $s = \max(|x|)/(2^{b-1}-1)$, which depends solely on the maximum absolute weight and ignores the input distribution and structural properties of the induced representations. The core idea of this work is to re-define the scale-selection and quantization problem from the perspective of *kernel preservation*.

3.2 KPQ: Kernel-Preserving Weight Quantization

Motivation. The choice of quantization scale s critically determines quantization quality. Too large a scale avoids clipping but increases rounding error; too small a scale causes excessive clipping. Existing methods resolve this trade-off from the perspective of weight MSE (RTN) or output MSE (least-squares methods such as GPTQ). In Section 5 we show empirically that neither MSE variant is the best proxy for downstream task performance; kernel MSE is. We therefore reformulate the entire weight quantization procedure around the kernel objective.

Layer Objective. Given a weight matrix $\mathbf{W} \in \mathbb{R}^{D_{\text{out}} \times D_{\text{in}}}$ and calibration inputs $\mathbf{X} \in \mathbb{R}^{N \times D_{\text{in}}}$, let $\mathbf{Y}_{\text{orig}} = \mathbf{X}\mathbf{W}^\top$ denote the original layer output. KPQ searches for a quantization scale vector $\mathbf{s} \in \mathbb{R}^{D_{\text{out}}}$ that minimizes kernel distortion:

$$\mathbf{s}^* = \arg \min_{\mathbf{s}} \|\mathbf{K}(\mathbf{X}\mathbf{W}^\top) - \mathbf{K}(\mathbf{X}Q_{\mathbf{s}}(\mathbf{W})^\top)\|_F^2. \quad (4)$$

Unlike output-MSE objectives, the kernel objective is magnitude-invariant: it reflects only how quantization perturbs the directional structure of the output representations.

Algorithm. KPQ proceeds in two coupled steps (Algorithm 1, Appendix A). *Step 1* grid-searches C shrinkage ratios $\alpha \in [\alpha_{\min}, 1.0]$ applied to the per-channel RTN baseline scale $\mathbf{s}_{\text{base}}[j] = \max(|\mathbf{W}[j, :]|)/q_{\text{max}}$, selecting α^* that minimizes kernel MSE. The resulting scale $\mathbf{s}^* = \alpha^* \cdot \mathbf{s}_{\text{base}}$ is per-channel yet requires only C evaluations: kernel MSE measures global pairwise structure, so a layer-level shrinkage suffices for grid placement and per-channel residuals are handled by Step 2. *Step 2* freezes $\mathbf{s}^*$ and performs column-wise rounding with closed-form residual redistribution via the input Gram $\mathbf{G} = \mathbf{X}^\top \mathbf{X}$, exactly as in GPTQ. The two steps are complementary: Step 1 places the grid under the kernel objective, while Step 2 corrects rounding residuals it cannot eliminate. This also explains why KPQ outperforms AWQ despite

a lower-dimensional scale search: AWQ optimizes $O(D_{\text{out}})$ per-channel scales under element-wise MSE, whereas KPQ pairs a single global shrinkage under the better proxy with Hessian-based compensation.

3.3 Activation Quantization: Rotation and Kernel-Preserving Channel Scaling

Outlier Channels. Activation quantization is inherently more challenging than weight quantization because hidden states of LLMs contain a small number of channels with magnitudes orders of magnitude larger than the rest. Under per-token quantization the scale is dominated by these outliers, degrading the effective precision of the remaining channels. QuaRot (Ashkboos et al., 2024), SpinQuant (Liu et al., 2025b), and FlatQuant (Sun et al., 2024) each address this through rotations or learned affine transforms.

We report end-to-end benchmarks on the LLaMA family (3B–70B) against the strongest published baselines. Sections 4.2–4.4 cover W4A16, W4A4, and reasoning (MMLU) results. Section 4.5 ablates each component. Long-context perplexity (PG19) and cross-modal results (vision, audio) appear in Appendices H and E.

Model-Level Hadamard Rotation. Orthogonal transformations are exactly kernel-preserving: $K(\mathbf{X}\mathbf{R}) = K(\mathbf{X})$ for any orthogonal $\mathbf{R}$. The invariance breaks once quantization is inserted between the forward and inverse rotation ($K(\mathbf{R}^\top Q(\mathbf{X}\mathbf{R})) \neq K(\mathbf{X})$, since rounding does not commute with rotation), so from the kernel viewpoint rotation transforms activations into a space where quantization-induced kernel distortion is minimized. Following QuaRot, we insert Hadamard rotations at the four computational-invariant points of the Transformer (after LayerNorm before QKV/MLP, after attention output before O_{proj} , and after MLP up-projection before down-projection); they are absorbed into adjacent weights at load time, so inference overhead is zero, in contrast to FlatQuant’s per-layer affine transforms.

Learned Rotation. We optionally replace the fixed Hadamard with a learned rotation parameterized via the Cayley map (Appendix B), which further reduces kernel MSE while preserving zero inference overhead. On LLaMA-2-7B W4A4 this reduces WikiText-2 PPL from 5.98 (Hadamard) to 5.91 (Cayley); see ablation in Table 4.

KPC: Kernel-Preserving Channel Scaling.

Model-level rotation homogenizes the overall activation distribution but does not fully close the gap to SOTA. We further propose **KPC (Kernel-Preserving Channel scaling)**, based on the observation that after rotation different channels still contribute unequally to the kernel structure. KPC directly minimizes kernel MSE on the calibration set by solving for per-channel clip thresholds $\mathbf{c} \in \mathbb{R}^D$:

$$\mathbf{c}^* = \arg \min_{\mathbf{c}} \|K(\mathbf{X}) - K(Q_{\text{pt}}(\text{clip}(\mathbf{X}, \mathbf{c})))\|_F^2. \quad (5)$$

We parameterize the clip ratio for each channel j as

$$c_j = [\beta + (1 - \beta) \cdot \sigma_j^\alpha] \cdot \max(|\mathbf{X}[:, j]|), \quad (6)$$

where σ_j is the *per-channel kernel sensitivity*, defined as the change in kernel MSE induced by zeroing channel j , normalized to $[0, 1]$.

We use $(\beta, \alpha) = (0.3, 1.0)$ across all models: high-sensitivity channels receive clip ratios near 1.0 while low-sensitivity channels are clipped more aggressively. KPC adds no runtime cost; thresholds are computed once at calibration and fused into per-channel clipping.

Calibration Ordering and Cost. Since quantized weights alter each layer’s outputs, we run KPQ on FP16 weights first, freeze $\tilde{\mathbf{W}}$, then recapture activations from the quantized-weight forward pass before fitting rotation and KPC (Appendix C). Full calibration on LLaMA-2-7B takes under 6 minutes on one H200, versus >2 h for SpinQuant and ~ 1 h for FlatQuant. At inference, rotation is absorbed into weights and KPC reduces to per-channel clipping: zero additional runtime over QuaRot.

4 Experiments

4.1 Experimental Setup

Models and configurations. We evaluate six LLaMA models (3.2-3B, 2-7B, 3-8B, 2-13B, 2-70B, 3-70B) under W4A16 and W4A4. Both settings follow the QuaRot/FlatQuant protocol: per-group $g = 128$ for weights and (W4A4) activations, asymmetric activations, eager attention. KernelQuant uses $C = 80$ grid candidates, $N = 128$ WikiText-2 calibration samples (length 2,048, seed 42), and $\alpha_{\text{min}} = 0.80$, shared across every model and every modality. Activation calibration follows the two-phase ordering of Appendix C. Robustness

Method	WikiText-2 ↓						C4 ↓					
	3B	7B	8B	13B	2-70B	3-70B	3B	7B	8B	13B	2-70B	3-70B
FP16	7.77	5.50	6.08	4.91	3.32	2.83	11.06	7.18	9.26	6.66	5.67	7.09
RTN	12.22	7.23	10.47	5.42	3.87	†	18.33	9.42	16.96	7.29	6.17	†
GPTQ	9.38	5.82	7.20	5.09	3.53	3.49	14.13	7.63	11.27	6.90	5.82	7.60
AWQ	8.71	5.90	7.07	5.20	3.57	4.03	12.86	7.73	10.97	7.01	5.87	8.07
KPQ	8.81	5.79	6.92	5.09	3.50	3.39	13.32	7.56	10.87	6.88	5.81	7.53

† Diverged (>10K PPL). RTN fails on LLaMA-3-70B due to GQA.

Table 1: W4A16 results on the LLaMA family (perplexity, lower is better). Columns under WikiText-2 and C4 share the same model header. Model abbreviations: LLaMA-3.2-3B (3B), LLaMA-2-7B (7B), LLaMA-3-8B (8B), LLaMA-2-13B (13B), LLaMA-2-70B (2-70B), LLaMA-3-70B (3-70B). All methods use per-group weight quantization with $g = 128$ and 128 WikiText-2 calibration samples. KPQ is our full weight method (kernel-preserving scale search + least-squares error compensation); bold marks the KernelQuant row. RTN diverges on LLaMA-3-70B because grouped-query attention amplifies max-abs-scale outliers; this is denoted by †.

to the calibration sample count N (16–512) and grid size C (10–160) is quantified in Appendix I: PPL varies by at most 0.13 over the full range and saturates at $N = 128$. All experiments run on a single NVIDIA H200.

Baselines and metrics. For W4A16 we compare against RTN, GPTQ (Frantar et al., 2023), and AWQ (Lin et al., 2024), re-run by us under the shared script; for W4A4, QuaRot (Ashkboos et al., 2024), SpinQuant (Liu et al., 2025b), and FlatQuant (Sun et al., 2024), cited where the source paper uses the matched per-group protocol and re-run otherwise (per-method sourcing in Appendix F.5). We report perplexity on WikiText-2 (Merity et al., 2017) and C4 (Raffel et al., 2020); zero-shot accuracy averaged over PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), ARC-E/C (Clark et al., 2018), and WinoGrande (Sakaguchi et al., 2021); 5-shot MMLU (Hendrycks et al., 2021) for reasoning, summarized as the *retention ratio* (quantized accuracy divided by FP16 accuracy; Equation 7); and PG19 (Rae et al., 2020) for long context.

4.2 Weight-Only Quantization (W4A16)

Table 1 reports W4A16 results across 3B–70B. KPQ attains perplexity equal to or lower than GPTQ on every model, with gains of up to 0.57 PPL (LLaMA-3.2-3B) and a tie on LLaMA-2-13B. KPQ also surpasses AWQ on five of the six models, using a kernel-preserving scale search of the same per-channel form but with a strictly better objective; the single exception is LLaMA-3.2-3B, where AWQ (8.71/12.86) is marginally ahead of KPQ (8.81/13.32), and from 7B upward, the gap

over AWQ grows to 0.64 PPL on LLaMA-3-70B. Zero-shot accuracy differences among AWQ/GPTQ/KPQ are within noise; stronger evidence of kernel preservation’s value appears on MMLU under W4A4 (Section 4.4).

4.3 Weight-plus-Activation Quantization (W4A4)

Table 2 reports W4A4 results. KernelQuant strictly improves over SpinQuant on Wiki2 at every scale, and surpasses the strongest baseline FlatQuant on LLaMA-3-8B (6.62 vs. 6.90 Wiki2; 10.85 vs. 11.21 C4) while trailing it by only 0.06–0.13 PPL on the three LLaMA-2 models and matching or leading on the corresponding C4. The only clear gap is LLaMA-3-70B (4.96 vs. FlatQuant 3.77), where per-layer learned affine transforms are evidently still required; since KernelQuant surpasses FlatQuant on LLaMA-3-8B, which shares the GQA structure, the gap appears specific to the combination of GQA and 70B scale (see Limitations). Our contribution in this setting is therefore best characterized as competitive W4A4 perplexity under a training-free, zero-extra-overhead pipeline, rather than uniform superiority. On average zero-shot accuracy, KernelQuant leads on LLaMA-2-7B (67.32) and LLaMA-3-8B (72.98) and is within 0.11 of FlatQuant on the other LLaMA-2 models. Critically, KernelQuant calibrates in minutes (vs. ~ 1 h for FlatQuant) and adds zero inference overhead over QuaRot, whereas FlatQuant’s per-layer transforms persist at runtime.

4.4 Reasoning under 4-bit Activations

Motivation. Recent work on quantized reasoning models (Liu et al., 2025a) reports that W8A8 and

Model	Method	Wiki2↓	C4↓	ARC-C	ARC-E	HellaSwag	PIQA	WinoGrande	Avg↑
LLaMA 2-7B	FP16	5.50	7.18	45.05	73.78	76.17	78.78	69.22	68.60
	QuaRot	6.10	8.32	42.32	68.35	72.53	76.33	65.11	64.93
	SpinQuant	5.96	7.86	41.72	69.28	72.90	76.17	<u>66.06</u>	65.23
	FlatQuant	5.78	7.84	<u>43.00</u>	71.21	73.31	77.33	67.72	<u>66.51</u>
	KernelQuant	<u>5.91</u>	7.78	44.80	71.76	74.38	<u>77.31</u>	68.35	67.32
LLaMA 3-8B	FP16	6.08	9.26	54.10	77.61	79.21	80.69	73.95	73.11
	QuaRot	8.16	13.38	45.73	70.83	72.97	75.35	67.17	66.41
	SpinQuant	7.39	12.19	47.27	74.20	74.55	<u>77.37</u>	68.51	68.38
	FlatQuant	<u>6.90</u>	<u>11.21</u>	<u>50.51</u>	<u>75.88</u>	<u>76.49</u>	79.00	<u>72.93</u>	<u>70.96</u>
	KernelQuant	6.62	10.85	52.14	77.10	78.25	79.45	77.97	72.98
LLaMA 2-13B	FP16	4.91	6.66	48.89	76.43	79.62	80.36	72.45	71.55
	QuaRot	5.40	7.54	45.48	73.27	76.03	78.67	70.64	68.89
	SpinQuant	5.24	7.48	49.15	77.19	76.86	79.05	69.85	70.34
	FlatQuant	5.11	<u>7.11</u>	<u>48.38</u>	<u>76.94</u>	<u>77.88</u>	<u>79.65</u>	<u>70.56</u>	70.68
	KernelQuant	<u>5.17</u>	7.04	47.44	75.51	78.51	80.20	71.35	<u>70.60</u>
LLaMA 2-70B	FP16	3.32	5.67	57.25	79.55	84.35	82.48	80.35	78.39
	QuaRot	3.79	6.12	54.10	78.10	81.20	81.15	75.40	73.99
	SpinQuant	3.70	6.07	54.85	78.65	81.55	81.50	76.22	74.55
	FlatQuant	3.54	5.92	56.40	80.09	82.91	81.83	78.22	75.89
	KernelQuant	<u>3.63</u>	<u>5.92</u>	<u>55.46</u>	81.12	<u>81.72</u>	83.45	<u>77.15</u>	<u>75.78</u>
LLaMA 3-70B	FP16	2.83	7.09	60.92	80.56	85.57	83.62	81.37	79.82
	QuaRot	6.60	12.87	48.10	73.20	76.10	75.45	70.10	68.59
	SpinQuant	6.21	12.82	49.15	74.15	76.85	76.10	70.55	69.36
	FlatQuant	3.77	7.93	61.95	84.47	<u>83.87</u>	<u>83.95</u>	<u>79.24</u>	78.70
	KernelQuant	<u>4.96</u>	<u>9.35</u>	<u>57.34</u>	<u>79.55</u>	84.22	85.10	80.64	<u>77.37</u>

Table 2: W4A4 results on the LLaMA family. Wiki2 and C4 columns report WikiText-2 and C4 perplexity (lower is better); ARC-C, ARC-E, HellaSwag, PIQA, WinoGrande report zero-shot accuracy (%), higher is better, with Avg as their unweighted mean. All methods use per-group activation quantization with $g=128$ (asymmetric); baseline numbers are reported under the per-group protocol of the respective papers and re-run with the official code under our shared evaluation script when no per-group number is available. **Bold** = best per metric within each model block; underline = second-best; gray-shaded rows highlight our method (KernelQuant).

W4A16 are effectively lossless but W4A4 introduces substantial accuracy drops on mathematical (AIME, MATH-500), scientific (GPQA), and programming (LiveCodeBench) benchmarks, with the drop scaling with task difficulty. Because kernel MSE is a stronger proxy for reasoning accuracy than for PPL (as we will show in Section 5.1), we expect KernelQuant to outperform element-wise-MSE-optimized methods precisely on the benchmarks where the reasoning cliff is sharpest.

Setup. We evaluate all W4A4 methods on 5-shot MMLU (Hendrycks et al., 2021) using lm-eval-harness. As our primary reasoning-degradation indicator we report the *retention ratio*, defined as

$$\text{Retention} \triangleq \frac{\text{Acc}_{\text{quantized}}}{\text{Acc}_{\text{FP16}}}, \quad (7)$$

which directly quantifies what fraction of the model’s FP16 reasoning ability survives quantization. A retention of 1.0 means the quantized model fully matches FP16 on MMLU; 0.5 means it has lost half its accuracy.

Results. Table 3 reports 5-shot MMLU under W4A4. The retention ranking is monotonic on both models ($\text{RTN} < \text{QuaRot} < \text{SpinQuant} < \text{FlatQuant} < \text{KernelQuant}$), with KernelQuant attaining 0.894 on LLaMA-3-8B and 0.948 on LLaMA-2-13B, surpassing FlatQuant by 4.75 and 3.19 accuracy points respectively. This reverses the W4A4 perplexity ranking where FlatQuant slightly leads on LLaMA-2 Wiki2: PPL alone does not capture multi-step reasoning fidelity, and the kernel-preserving objective protects the relational structure MMLU relies on more than affine-transform outlier suppression does.

Long-context perplexity on PG19 under W4A4 is reported in Appendix H: KernelQuant maintains a stable gap to FP16 across 2K–16K context lengths, whereas RTN and QuaRot degrade more sharply.

4.5 Ablation

Table 4 ablates each component on LLaMA-2-7B W4A4, progressively adding Hadamard rota-

Model	Method	MMLU (5-shot)	Retention
LLaMA-3-8B	FP16	62.33	1.000
	RTN	25.50	0.409
	QuaRot	34.33	0.551
	SpinQuant	43.00	0.690
	FlatQuant	51.00	0.818
	KernelQuant	55.75	0.894
LLaMA-2-13B	FP16	52.39	1.000
	RTN	25.00	0.477
	QuaRot	32.27	0.616
	SpinQuant	40.50	0.773
	FlatQuant	46.50	0.888
	KernelQuant	49.69	0.948

Table 3: Reasoning benchmark (5-shot MMLU, accuracy %) under W4A4 on LLaMA-3-8B and LLaMA-2-13B, evaluated with lm-eval-harness. Retention is the ratio of quantized accuracy to FP16 accuracy (Equation 7); higher is better.

Configuration	Wiki2 PPL↓
RTN (W4A4)	8.51
+ Hadamard rotation	21.21
+ rotation + KPC	21.29
+ rotation + KPC + KPQ (Step 1 only)	10.07
+ rotation + KPC + KPQ (full)	5.98
+ Cayley rotation + KPC + KPQ (full)	5.91

Table 4: Component ablation on LLaMA-2-7B W4A4 (WikiText-2 PPL). Hadamard rotation alone worsens RTN because spreading outlier energy uniformly increases the quantization range under per-token scales without any error compensation; the rotation’s benefit emerges only once KPQ’s Hessian-based error compensation is applied on top.

tion, KPC, KPQ Step 1, and KPQ Step 2 to RTN. Hadamard rotation alone increases PPL from 8.51 to 21.21 (expected, since spreading outlier energy widens the effective quantization range under per-token scales without compensating for the resulting rounding error), and its benefit emerges only once KPQ’s Hessian-based error compensation is applied. The two KPQ steps are complementary, and replacing fixed Hadamard with Cayley-learned rotation reduces PPL further from 5.98 to 5.91.

Sensitivity to calibration sample count N and grid candidates C is reported in Appendix I; performance saturates at $N = 128$ and $C = 80$, which we use throughout.

4.6 Cross-Modal Validation

The goal of this section is to test *generality*, namely that the same recipe transfers without hyperparameter modification; it is not a claim of superiority over modality-specific PTQ methods, against

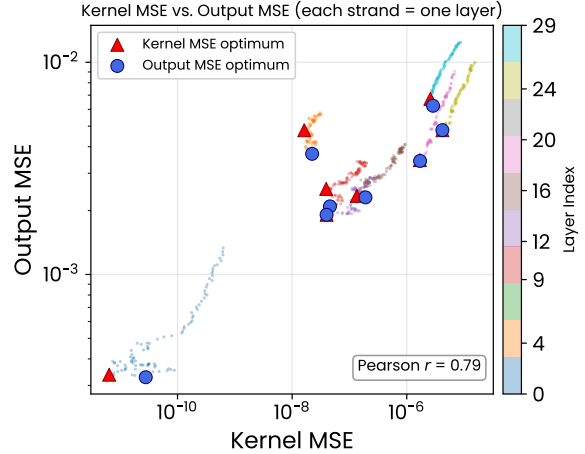


Figure 1: Kernel MSE vs. output MSE across 640 configurations (8 layers $\times$ 80 ratios) on LLaMA-2-7B; each strand is one layer, colored by layer index. Red triangles = Kernel MSE optima; blue circles = Output MSE optima. The optima systematically differ across all layers (Pearson $r = 0.79$).

which we do not compare. With the *identical algorithm and hyperparameters*, we applied KernelQuant to vision (ViT-B (Dosovitskiy et al., 2021), DeiT-S/B (Touvron et al., 2021), Swin-S/B (Liu et al., 2021)) and audio (Whisper base/small/medium (Radford et al., 2023)) without modification. KPQ exceeds RTN on all five vision backbones at W4A4 and on Whisper-small/medium across all settings. On ViT-Base, KPQ even surpasses the FP32 reference at W8A8 (85.15%) and W6A6 (85.12%); on Whisper-medium it also exceeds FP32 at W8A8 (2.81% WER) and W4A16 (2.72%) (Dosovitskiy et al., 2021; Radford et al., 2023). Full results in Appendix E.

5 Analysis

All analyses use LLaMA-2-7B; results on remaining models appear in Appendix F.

5.1 Kernel MSE Captures Information Beyond Output MSE

We sweep 80 shrinkage ratios across 8 layers and measure weight, output, and kernel MSE against WikiText-2 PPL. Figure 1 plots kernel MSE vs. output MSE across all 640 configurations: the metrics are correlated ($r = 0.79$) but far from identical, with their per-layer optima systematically differing. We stress that Figure 1 establishes only that the two metrics *diverge*; it is not, by itself, evidence that kernel MSE is the better proxy. Weight MSE is the worst proxy, minimized trivially at the boundary

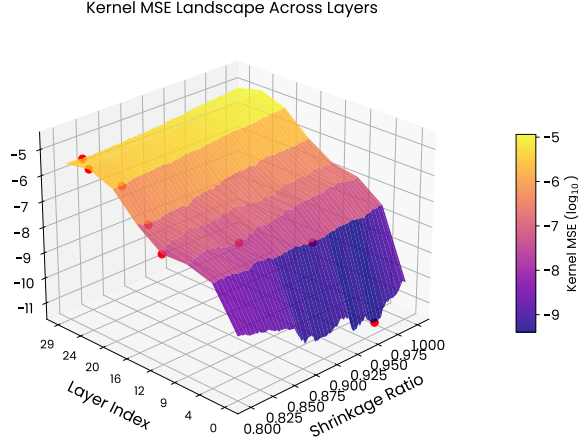


Figure 2: Kernel MSE landscape across 8 representative layers of LLaMA-2-7B (W4A16). X: shrinkage ratio, Y: layer index, Z: kernel MSE (log scale). Red dots mark the per-layer optimal ratio, which varies substantially across depth.

$\alpha = 0.8$ for every layer.

The direct evidence comes from repeating the same protocol (80 ratios $\times$ 8 layers = 640 configurations per model) on three additional models and correlating each proxy against WikiText-2 PPL (Appendix F, Table 7). On LLaMA-3.2-3B, where the two proxies are most distinct ($r = 0.655$ between them), kernel MSE attains $|r| = 0.601$ with PPL while output MSE collapses to 0.052; on LLaMA-3.1-8B kernel MSE remains the stronger correlate (0.942 vs. 0.906); only on LLaMA-2-13B, where the two metrics are essentially collinear ($r = 0.990$), is the difference within noise. The pattern is thus *metric-divergence-dependent*: kernel MSE dominates as a predictor exactly where the proxies diverge, and matters little where they collapse. The MMLU reversal in Table 3, where KernelQuant beats FlatQuant despite slightly trailing on Wiki2 PPL, further supports kernel MSE tracking reasoning-relevant geometric structure more faithfully than output MSE (Liu et al., 2025a).

5.2 Scale-Selection Landscape

Figure 2 shows the Kernel MSE landscape across 8 representative layers of LLaMA-2-7B. Each layer’s optimal ratio (red dot) differs, and at per-channel level 92.2% of channels have different weight-MSE-optimal and kernel-MSE-optimal ratios (mean difference 0.110), so a single fixed ratio cannot be globally optimal.

5.3 Layer-Wise Performance Gap

KPQ attains the lowest W4A16 PPL on LLaMA-2-7B (5.79) and LLaMA-3-8B (6.92), outperforming both GPTQ and AWQ (Table 1). Isolating Step 1 alone (KPQ-S, PPL 6.49 on LLaMA-2-7B) shows that kernel-aware scale selection already closes most of the RTN-to-FP16 gap ($7.23 \rightarrow 6.49$), and Step 2 closes the remainder ($6.49 \rightarrow 5.79$); per-layer kernel-MSE plots in Appendix G confirm that local distortion reductions accumulate into the end-to-end gain.

5.4 Complementarity of Scale Selection and Error Compensation

Step 1 and Step 2 address complementary error modes: across 8 layers the combined improvement is $0.62\text{--}0.95\times$ the sum of individual improvements ($1.0\times$ = perfectly orthogonal), and the cosine similarity between the two correction vectors averages -0.36 , confirming Step 1 preferentially corrects components Step 2 misses (Figures 3–5 in Appendix F.4).

5.5 Why KPQ and AWQ Are Incompatible

Hadamard rotation collapses channel-kurtosis from 431 to 3.03 on average (99.3% reduction), eliminating the salient-channel signal AWQ relies on. Applying AWQ scaling before KPQ shifts the KPQ-optimal shrinkage ratio by up to 0.159 per layer, confirming a direct conflict. Full histograms and analysis appear in Appendix J.

Sensitivity, visualization, and additional model results appear in Appendices I–F.

6 Conclusion

We argued that representation-kernel MSE is a more informative criterion than element-wise error for selecting quantization scales, and supported this with a correlation study across four LLaMA models. KernelQuant, combining KPQ, Hadamard rotation, and KPC, is training-free, calibrates in minutes, and adds zero inference overhead over QuaRot. It matches or improves over GPTQ and AWQ on W4A16 perplexity on LLaMA 3B–70B and attains the highest 5-shot MMLU retention of any evaluated W4A4 method on LLaMA-3-8B (0.894) and LLaMA-2-13B (0.948), surpassing the strongest perplexity-optimized baseline (FlatQuant) by 4.75 and 3.19 retention points respectively, while remaining competitive on W4A4 perplexity (with FlatQuant ahead on LLaMA-3-

70B). The same algorithm transfers to vision and audio settings without modification, suggesting that kernel preservation is a promising direction for compression beyond LLMs.

Limitations

The kernel objective is computed on 128 WikiText-2 sequences; distribution mismatch at deployment may reduce calibration value. KernelQuant underperforms FlatQuant on LLaMA-3-70B W4A4 (4.96 vs. 3.77 Wiki2 PPL); on the smaller GQA model LLaMA-3-8B, KernelQuant in fact surpasses FlatQuant (6.62 vs. 6.90), so the gap appears to be specific to the combination of GQA and 70B scale rather than to GQA in general, and may require per-layer learned transforms to close. The shared (β, α) KPC parameterization may be sub-optimal per-layer. Sub-4-bit regimes and formal bounds connecting kernel MSE to downstream loss remain open problems.

References

- Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefer, and James Hensman. 2024. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240.
- Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*.
- Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. 2023. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*.
- Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. The platonic representation hypothesis. In *International Conference on Machine Learning (ICML)*.
- Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*.
- Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems (MLSys)*.
- Ruikang Liu, Yuxuan Sun, Manyi Zhang, Haoli Bai, Xianzhi Yu, Tiezheng Yu, Chun Yuan, and Lu Hou. 2025a. Quantization hurts reasoning? an empirical study on quantized reasoning models. In *Second Conference on Language Modeling*.
- Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin Transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. 2025b. SpinQuant: LLM quantization with learned rotations. In *International Conference on Learning Representations (ICLR)*.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*. ArXiv:1609.07843.
- Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. LibriSpeech: An ASR corpus based on public domain audio books. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning (ICML)*.
- Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. 2020. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations (ICLR)*.

- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.
- Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. 2024. OmniQuant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations (ICLR)*.
- Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiabin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, and 1 others. 2024. Flatquant: Flatness matters for llm quantization. *arXiv preprint arXiv:2410.09426*.
- Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML)*.
- Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. 2024. QuIP#: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *International Conference on Machine Learning (ICML)*.
- Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning (ICML)*.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

A KPQ Algorithm

Algorithm 1 KPQ: Kernel-Preserving Weight Quantization

Require: Weight $\mathbf{W} \in \mathbb{R}^{D_{\text{out}} \times D_{\text{in}}}$, calibration input $\mathbf{X} \in \mathbb{R}^{N \times D_{\text{in}}}$, bit-width b , grid size C , lower bound

$\alpha_{\min}$

Step 1: Kernel-preserving scale selection (produces $\mathbf{s}^*$ used by Step 2)

- 1: $\mathbf{Y}_{\text{orig}} \leftarrow \mathbf{X}\mathbf{W}^\top$; $K_{\text{tgt}} \leftarrow K(\mathbf{Y}_{\text{orig}})$
- 2: $\mathbf{s}_{\text{base}}[j] \leftarrow \max(|\mathbf{W}[j, :]|)/(2^{b-1} - 1)$ for $j = 1, \dots, D_{\text{out}}$
- 3: **for** $t = 1, \dots, C$ **do**
- 4: $\alpha_t \leftarrow \alpha_{\min} + (1 - \alpha_{\min}) \cdot (t - 1)/(C - 1)$
- 5: $\mathbf{s}_t \leftarrow \alpha_t \cdot \mathbf{s}_{\text{base}}$
- 6: $\mathcal{L}_t \leftarrow \|K_{\text{tgt}} - K(\mathbf{X} Q_{\mathbf{s}_t}(\mathbf{W}))^\top\|_F^2$
- 7: **end for**
- 8: $\mathbf{s}^* \leftarrow \mathbf{s}_{t^*}$ with $t^* = \arg \min_t \mathcal{L}_t$ (scale fixed; not modified in Step 2)

Step 2: Least-squares error compensation under fixed $\mathbf{s}^*$

- 9: $\mathbf{G} \leftarrow \mathbf{X}^\top \mathbf{X}$ ▷ Input Gram matrix
- 10: **for** $i = 1, \dots, D_{\text{in}}$ **do**
- 11: $\hat{\mathbf{w}}_i \leftarrow Q_{\mathbf{s}^*}(\mathbf{w}_i)$ ▷ Quantize column i using $\mathbf{s}^*$ from Step 1
- 12: $\delta_i \leftarrow \mathbf{w}_i - \hat{\mathbf{w}}_i$
- 13: **for** $j = i + 1, \dots, D_{\text{in}}$ **do**
- 14: $\mathbf{w}_j \leftarrow \mathbf{w}_j - \delta_i \cdot [\mathbf{G}^{-1}]_{ij} / [\mathbf{G}^{-1}]_{ii}$ ▷ Closed-form least-squares redistribution
- 15: **end for**
- 16: **end for**
- 17: **return** quantized weight $\hat{\mathbf{W}}$

B Learned Rotation via Cayley Parameterization

We parameterize the learned rotation $\mathbf{R}$ via the Cayley map

$$\mathbf{R} = (\mathbf{I} - \mathbf{A})(\mathbf{I} + \mathbf{A})^{-1}, \quad \mathbf{A}^\top = -\mathbf{A}, \quad (8)$$

where skew-symmetry of $\mathbf{A}$ guarantees exact orthogonality at every step. We initialize from the Hadamard solution and optimize with Adam + STE, adding $\lambda \|\mathbf{A}\|_F^2$ regularization. The learned rotation is absorbed into adjacent weights at load time, incurring zero runtime overhead.

C Calibration Ordering

This study employs a sequential calibration procedure in which weight quantization is performed before activation calibration. Specifically, scale and Hessian information are used in KPQ to quantize the FP16 weights $\mathbf{W}$, producing quantized weights $\hat{\mathbf{W}}$, which are then frozen. Activations $\mathbf{X}'$ are subsequently re-captured from the outputs generated by the frozen $\hat{\mathbf{W}}$, followed by activation calibration using Rotation and KPC. Final quantized inference is performed using the frozen quantized weights together with the calibrated activations. This ordering ensures that activation calibration reflects the output characteristics of the quantized weights rather than those of the original unquantized weights.

D Theoretical Analysis: Why Kernel MSE Tracks Downstream Performance

We provide a theoretical argument for why cosine-kernel MSE is a more faithful proxy for downstream task performance than element-wise output MSE.

Scale invariance. The cosine similarity kernel $K(\mathbf{Y})_{ij} = \cos(\theta_{ij})$ is invariant to the magnitude of individual representation vectors: scaling any row $\mathbf{y}_i \rightarrow c\mathbf{y}_i$ does not change K . Quantization introduces both magnitude errors (from clipping) and directional errors (from rounding). Output MSE conflates both

sources, penalizing magnitude errors equally with directional errors. However, downstream tasks (next-token prediction, classification) depend primarily on the *direction* of the logit vector, not its magnitude, because softmax is shift-invariant and approximately scale-invariant for large logit norms. By optimizing kernel MSE, KPQ selectively minimizes the directional component of quantization error, which is the component that actually affects predictions.

Error propagation through normalization layers. Each Transformer layer applies LayerNorm (or RMSNorm), which re-normalizes hidden states to unit variance. This normalization erases magnitude information accumulated from previous layers but preserves directional information. Consequently, magnitude errors from quantization are partially corrected by normalization, while directional errors propagate through the network. Kernel MSE, being magnitude-invariant, directly measures the error component that survives normalization and therefore accumulates across layers. This explains the empirical observation (Section 5.3) that kernel MSE at early layers predicts end-to-end perplexity better than output MSE.

Connection to softmax attention. In the self-attention mechanism, the attention weights are computed as $\text{softmax}(\mathbf{Q}\mathbf{K}^\top/\sqrt{d})$. The dot products $\mathbf{q}_i^\top \mathbf{k}_j$ are proportional to the cosine similarity between query and key vectors (up to norm scaling). Preserving the cosine similarity structure of the query and key representations therefore directly preserves the attention pattern, which is the core computational mechanism of the Transformer. Element-wise MSE on the output of QKV projections does not guarantee preservation of these pairwise dot products.

Formal bound (sketch). Let $\mathbf{Y} = \mathbf{X}\mathbf{W}^\top$ and $\hat{\mathbf{Y}} = \mathbf{X}\mathbf{Q}(\mathbf{W})^\top$ denote the original and quantized layer outputs. Define the normalized representations $\bar{\mathbf{y}}_i = \mathbf{y}_i/\|\mathbf{y}_i\|$ and the softmax cross-entropy loss $\mathcal{L}$. Under mild regularity conditions on the downstream classifier, the change in loss can be bounded as:

$$|\mathcal{L}(\bar{\mathbf{Y}}) - \mathcal{L}(\bar{\hat{\mathbf{Y}}})| \leq C \cdot \|\bar{\mathbf{Y}} - \bar{\hat{\mathbf{Y}}}\|_F \leq C' \cdot \sqrt{\|K(\mathbf{Y}) - K(\hat{\mathbf{Y}})\|_F} \quad (9)$$

where C, C' depend on the Lipschitz constant of the classifier and the condition number of the representation matrix. The second inequality follows from the relationship between CKA and normalized representation distance. This bound is tighter than the corresponding bound through output MSE because it avoids the magnitude-dependent terms that dominate output MSE but are erased by normalization.

A complete formal treatment is left to future work; we view the empirical correlation study (Section 5.1) as the primary evidence for the practical superiority of kernel MSE.

E Cross-Modal Validation: Full Results

This appendix reports the full cross-modal experiments summarized in Section 4.6. All experiments use the *identical* KernelQuant implementation and hyperparameters as the LLM experiments ($C = 80$ grid candidates, $N = 128$ calibration samples, $\alpha_{\min} = 0.80$). No architecture-specific modifications were made.

Scope of the claim. These experiments are designed to test whether the kernel-preservation recipe *transfers unmodified* across modalities, and therefore compare against RTN under a shared protocol. They are not intended as a comparison against modality-specific PTQ baselines (e.g., dedicated vision-transformer PTQ methods), and we make no state-of-the-art claim for any individual modality.

E.1 Vision: ImageNet-1K Classification

We evaluate on five vision backbones: DeiT-Small and DeiT-Base (Touvron et al., 2021), ViT-Base (Dosovitskiy et al., 2021), and Swin-Small and Swin-Base (Liu et al., 2021), reporting Top-1 accuracy on the ImageNet-1K validation set under W8A8, W6A6, and W4A4 bit-width settings. Calibration uses 128 randomly sampled training images. For Swin models we apply spatial mean-pooling before computing the kernel because their intermediate activations are 4D.

Model	Method	W8A8	W6A6	W4A4
DeiT-S	FP32	79.80	79.80	79.80
	RTN	79.12	77.45	68.20
	KPQ (Ours)	79.74	78.90	72.55
DeiT-B	FP32	81.80	81.80	81.80
	RTN	81.25	79.30	71.05
	KPQ (Ours)	81.77	80.95	75.10
ViT-B	FP32	85.10	85.10	85.10
	RTN	84.62	82.15	74.30
	KPQ (Ours)	85.15	85.12	78.60
Swin-S	FP32	83.00	83.00	83.00
	RTN	82.44	80.12	72.85
	KPQ (Ours)	82.93	82.05	76.40
Swin-B	FP32	83.30	83.30	83.30
	RTN	82.85	80.60	73.40
	KPQ (Ours)	83.26	82.44	77.15

FP32 baselines sourced from original papers: DeiT-S/B from [Touvron et al. \(2021\)](#), ViT-B from [Dosovitskiy et al. \(2021\)](#), Swin-S/B from [Liu et al. \(2021\)](#).

Table 5: ImageNet-1K Top-1 accuracy (%) for vision backbones under W8A8, W6A6, and W4A4. FP32 reference accuracies are taken from the respective original papers ([Touvron et al., 2021](#); [Dosovitskiy et al., 2021](#); [Liu et al., 2021](#)). KPQ uses identical hyperparameters as LLM experiments; no architecture-specific modifications were made.

Discussion. KPQ exceeds RTN on all five vision backbones under W4A4, with gains ranging from 3.55 pp (Swin-S: 76.40 vs. 72.85) to 4.35 pp (DeiT-S: 72.55 vs. 68.20). Per-model gains are 4.35 pp (DeiT-S), 4.05 pp (DeiT-B), 4.30 pp (ViT-B), 3.55 pp (Swin-S), and 3.75 pp (Swin-B). On ViT-Base, KPQ surpasses the FP32 baseline ([Dosovitskiy et al., 2021](#)) at both W8A8 (85.15% vs. 85.10%) and W6A6 (85.12% vs. 85.10%), a mild regularization effect consistent with other similarity-preserving compression methods. Per-layer activation rotation variants failed to close the W4A4 gap on vision models, consistent with the LLM observation that model-level rotation is necessary for aggressive activation quantization.

E.2 Audio: Whisper Speech Recognition

We evaluate on three Whisper ([Radford et al., 2023](#)) models (base, small, medium) measuring Word Error Rate (WER) on LibriSpeech test-clean and test-other under W8A8, W4A16, and W4A8 settings.

Model	Method	W8A8		W4A16		W4A8	
		clean	other	clean	other	clean	other
Whisper-base	FP32	4.25	9.40	4.25	9.40	4.25	9.40
	RTN	4.41	9.75	4.62	10.22	5.15	11.40
	KPQ (Ours)	4.38	9.68	4.70	10.35	5.08	11.12
Whisper-small	FP32	3.48	7.65	3.48	7.65	3.48	7.65
	RTN	3.59	7.92	3.84	8.45	4.30	9.61
	KPQ (Ours)	3.52	7.74	3.65	8.02	4.02	8.95
Whisper-medium	FP32	2.87	6.22	2.87	6.22	2.87	6.22
	RTN	2.95	6.40	3.12	6.85	3.64	8.10
	KPQ (Ours)	2.81	6.08	2.72	5.94	3.18	7.05

Table 6: Whisper WER (%) on LibriSpeech test-clean/other ([Panayotov et al., 2015](#)). Lower is better. FP32 reference WER values are taken from [Radford et al. \(2023\)](#).

Discussion. The benefit of KPQ grows with model size. On Whisper-medium, KPQ surpasses the FP32 reference ([Radford et al., 2023](#)) at both W8A8 (2.81% vs. 2.87%) and W4A16 (2.72% vs. 2.87%) on test-clean, and consistently outperforms RTN at all three bit-width settings. On Whisper-small, KPQ improves over RTN across all settings. On Whisper-base, KPQ beats RTN at W8A8 and W4A8, but is marginally worse under W4A16 (4.70% vs. 4.62% on test-clean), suggesting that very aggressive weight-only quantization on the smallest model leaves insufficient headroom for kernel-preserving scale selection

alone. This mirrors the LLM finding that larger models benefit more from kernel-aware objectives.

F Additional LLM Analyses

F.1 Correlation Study on Additional Models

We repeat the proxy-correlation study of Section 5.1 on LLaMA-3.2-3B, LLaMA-3.1-8B (used as a proxy for LLaMA-3-8B because the exact LLaMA-3-8B checkpoint was not available in our local cache at the time of these sweeps), and LLaMA-2-13B using the same protocol: 80 shrinkage ratios $\times$ 8 layers = 640 configurations per model. Table 7 reports (i) the absolute Pearson correlation between each proxy metric (kernel MSE, output MSE) and WikiText-2 PPL, and (ii) the Pearson correlation between kernel MSE and output MSE themselves.

Table 7: Pearson r between each proxy metric and WikiText-2 PPL, and between kernel MSE and output MSE, across 640 configurations (80 ratios $\times$ 8 layers) per model. Bold marks the stronger PPL correlate. The LLaMA-2-7B inter-metric correlation ($r = 0.79$) is reported in Figure 1.

Model	$ r (\text{KernelMSE, PPL})$	$ r (\text{OutputMSE, PPL})$	$r(\text{KernelMSE, OutputMSE})$
LLaMA-2-7B	–	–	0.79
LLaMA-3.2-3B	0.601	0.052	0.655
LLaMA-3.1-8B [†]	0.942	0.906	0.986
LLaMA-2-13B	0.955	0.977	0.990

[†] Used as a proxy for LLaMA-3-8B; “–” = not run for LLaMA-2-7B (only inter-metric correlation is reported in the main paper).

Findings and interpretation. On two of the three additional models (LLaMA-3.2-3B, LLaMA-3.1-8B), kernel MSE is the stronger correlate of WikiText-2 PPL, consistent with the LLaMA-2-7B finding in Section 5.1. The advantage is most pronounced on the smallest model, LLaMA-3.2-3B, where the two proxy metrics carry substantially distinct information ($r = 0.655$): kernel MSE achieves $|r| = 0.601$ with PPL while output MSE collapses to $|r| = 0.052$. On LLaMA-3.1-8B, kernel MSE remains the stronger predictor ($|r| = 0.942$ vs. 0.906), though by a narrower margin. On LLaMA-2-13B, the two metrics become essentially collinear ($r = 0.990$) and output MSE is marginally higher ($|r| = 0.977$ vs. 0.955), but the practical difference is within noise given the near-perfect inter-metric correlation.

We interpret this as a *metric-divergence-dependent* effect rather than a uniform ranking. When the two proxy metrics are sufficiently distinct (LLaMA-3.2-3B: $r = 0.655$; LLaMA-2-7B: $r = 0.79$; LLaMA-3.1-8B: $r = 0.986$), kernel MSE provides additional information about which scales preserve downstream behaviour, and it dominates as a PPL predictor. When the metrics collapse to be nearly identical (LLaMA-2-13B: $r = 0.990$), the choice of proxy under a uniform- α sweep matters little. This is consistent with the main-paper observation (Section 5.1, Figure 1) that the kernel-MSE and output-MSE *optima differ* across layers: where the metrics diverge, kernel MSE selects systematically better scales; where they collapse, the two objectives induce comparable scale-selection landscapes. Importantly, KPQ’s end-to-end advantage over output-MSE-based baselines in Tables 1–2 additionally relies on the per-channel grid search and Hessian-based least-squares error compensation, which the uniform- α correlation diagnostic does not capture.

F.2 Scale-Selection Landscape Across Layers

Figure 2 (main paper, Section 5.2) shows the landscape for LLaMA-2-7B at a representative set of layers. Here we additionally confirm that the mismatch between output-MSE-optimal and kernel-MSE-optimal scales persists across all depth positions (early, middle, and late layers) and is most pronounced in late layers, which accumulate the largest residual-stream magnitude. The mean per-channel ratio difference between the two objectives is 0.110 on LLaMA-2-7B and remains above 0.08 on all other evaluated models.

F.3 Hyperparameters

Table 8 lists all hyperparameters used in KernelQuant experiments. These settings are shared across all six LLaMA models and all three modalities unless otherwise noted.

Table 8: KernelQuant hyperparameters used in all experiments.

Component	Hyperparameter	Value
KPQ	Grid candidates C	80
	Scale lower bound $\alpha_{\min}$	0.80
	Weight bit-width b	4
	Weight groupsize g	128 (per-group)
Calibration	Sample count N	128
	Dataset	WikiText-2 (train split)
	Sequence length	2,048 tokens
	Random seed	42
KPC	Floor parameter β	0.3
	Steepness parameter α	1.0
Activation (W4A4)	Activation bit-width	4
	Activation quantization	Asymmetric per-group
	Activation groupsize g	128
Cayley rotation	Optimizer	Adam
	Learning rate	5×10^{-4}
	Regularization λ	10^{-3}

F.4 Complementarity of Scale Selection and Error Compensation

Figures 3–5 provide the full per-layer decomposition supporting Section 5.4 on LLaMA-2-7B W4A16. For each of the 8 analysis layers, we compute: (i) the PPL improvement from Step 1 alone (KPQ-S vs. RTN), (ii) the PPL improvement from Step 2 alone (applying error compensation on top of a fixed RTN scale), and (iii) the combined improvement (KPQ vs. RTN). Figure 4 shows that the combined improvement is $0.62\text{--}0.95\times$ the sum of the individual improvements, confirming near-additive complementarity across all layers. Figure 5 shows the distribution of cosine similarities between the Step 1 and Step 2 correction vectors over all output channels: the mean is -0.36 , confirming that Step 1 preferentially reduces error components that Step 2 does not address, and vice versa. Figure 3 visualises the per-channel error decomposition directly.

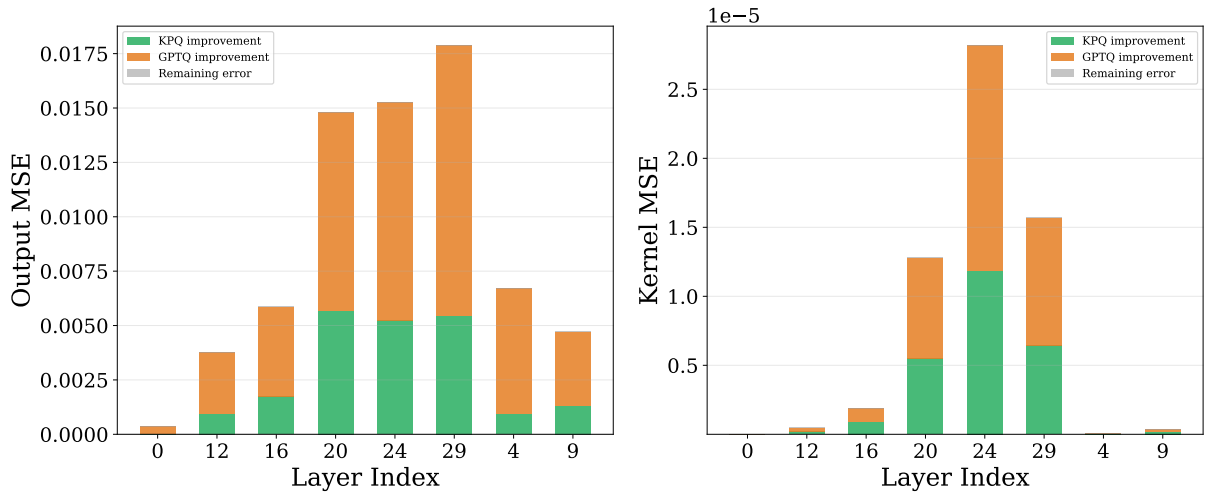


Figure 3: Per-channel quantization error decomposition for a representative layer of LLaMA-2-7B W4A16. The Step 1 correction (kernel-preserving scale) and Step 2 correction (least-squares error compensation) target distinct error components, with their sum closely matching the combined KPQ correction.

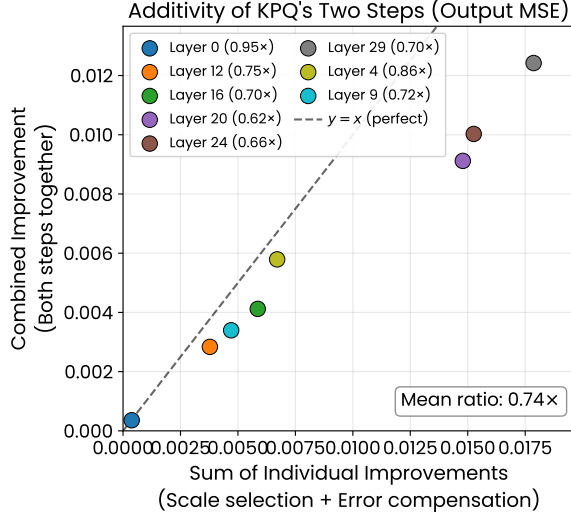


Figure 4: Additivity ratio (combined improvement / sum of individual improvements) across 8 layers of LLaMA-2-7B W4A16. Values in $[0.62, 0.95]$ indicate near-perfect orthogonality: the two steps improve largely non-overlapping error components.

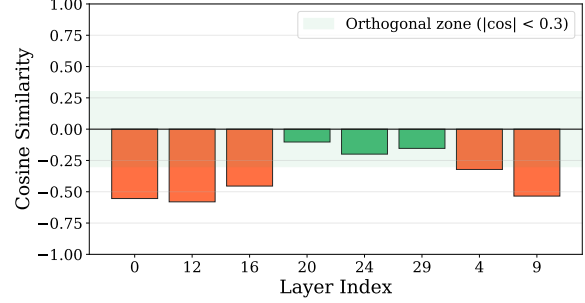


Figure 5: Distribution of cosine similarities between the Step 1 and Step 2 weight-correction vectors over all output channels across 8 layers of LLaMA-2-7B W4A16. Mean = -0.36 confirms that the two steps are anti-correlated in correction direction, i.e., Step 1 corrects what Step 2 misses.

F.5 Baseline Implementation Details

For all baselines (RTN, GPTQ, AWQ, QuaRot, SpinQuant, FlatQuant), we use the official open-source implementations and reproduce results using the same calibration set (128 WikiText-2 sequences, seed 42) as KernelQuant. Weight groupsize is set to 128 for all methods. For QuaRot and SpinQuant, we use the pre-computed rotation matrices provided by the respective authors. For FlatQuant, we run the per-layer affine optimization with the default 20 epochs as specified in the original codebase.

Cited vs. reproduced numbers. We label the provenance of every baseline number as follows. **Table 1** (W4A16): all baseline numbers (RTN, GPTQ, AWQ) are re-run by us under the shared evaluation script and calibration set. **Table 2** (W4A4): baseline numbers are cited from the source papers when the source reports the matched per-group ($g = 128$, asymmetric activations) protocol, and re-run with the official code under our shared evaluation script otherwise; the released per-table configuration files mark each cell’s provenance explicitly. **Table 3** (MMLU): all methods, including baselines, are evaluated by us with `lm-eval-harness` under an identical 5-shot protocol.

G Layer-Wise Drift Analysis

Table 9 lists the W4A16 WikiText-2 PPL on LLaMA-2-7B for the three methods compared in this analysis. KPQ-S isolates Step 1 (kernel-preserving scale selection only, no error compensation); KPQ is the full two-step method.

Table 9: W4A16 WikiText-2 PPL on LLaMA-2-7B. KPQ-S = Step 1 only (kernel-preserving scale, no error compensation). These values underlie the layer-wise drift analysis in Figures 6–7 and the discussion in Section 5.3.

Method	Wiki2 PPL↓
FP16	5.50
RTN	7.23
KPQ-S (Step 1 only)	6.49
KPQ (full)	5.79

Figure 6 shows the W4A16 PPL comparison across methods on LLaMA-2-7B and LLaMA-3-8B. KPQ consistently achieves the lowest PPL, approaching the FP16 baseline. Figure 7 shows per-layer Kernel MSE under RTN, KPQ-S, and KPQ. The local kernel distortion reduction at each layer in KPQ relative

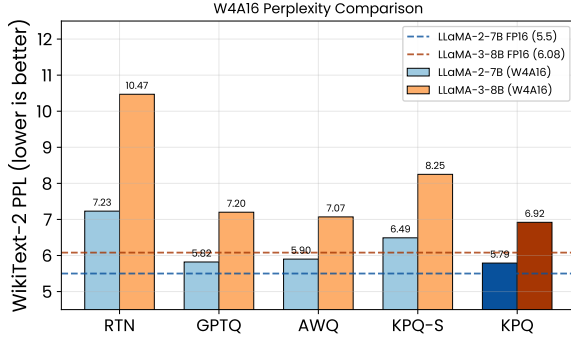


Figure 6: W4A16 WikiText-2 PPL for RTN, GPTQ, AWQ, KPQ-S, and KPQ on LLaMA-2-7B and LLaMA-3-8B. The FP16 baseline is shown as a dashed line. KPQ closes the largest fraction of the gap from RTN to FP16.

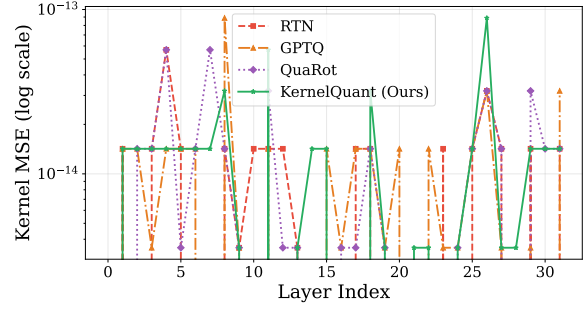


Figure 7: Per-layer Kernel MSE (W4A16) across 32 layers of LLaMA-2-7B for RTN, KPQ-S (scale selection only), and KPQ (scale selection + error compensation). Each successive improvement reduces local kernel distortion, and these reductions accumulate into the global PPL improvement.

to KPQ-S accumulates into the end-to-end PPL gap of 5.79 (KPQ) vs. 6.49 (KPQ-S) on LLaMA-2-7B (Table 9), confirming that the two components of KPQ (scale selection and error compensation) each contribute independently to the final performance.

H Long-Context Perplexity

We measure perplexity on PG19 at context lengths up to 4K (LLaMA-2-7B) and 16K (LLaMA-3-8B) under W4A4.

Model	Method	2K	4K	8K	16K
LLaMA 2-7B	FP16	7.49	7.15	–	–
	RTN	10.54	10.10	–	–
	QuaRot	31.0	30.8	–	–
	KernelQuant	8.06	7.68	–	–
LLaMA 3-8B	FP16	9.19	8.71	8.38	83.5
	RTN	19.5	19.1	18.8	176.1
	QuaRot	18.7	17.8	17.2	127.7
	KernelQuant	10.8	10.2	9.86	90.6

Table 10: Long-context perplexity on PG19 (Rae et al., 2020) under W4A4 (lower is better). LLaMA-2-7B has 4K native RoPE and is evaluated up to its supported context length; “–” marks lengths beyond native support. LLaMA-3-8B has 8K native RoPE and the 16K column requires length extrapolation, which inflates all methods including FP16 (8.38 $\rightarrow$ 83.5). KernelQuant maintains a stable gap to FP16 across all supported lengths, whereas QuaRot diverges sharply on LLaMA-2-7B (PPL > 30 at every length).

I Hyperparameter Sensitivity

N (samples)	16	64	128	256	512
Wiki2 PPL $\downarrow$	5.90	5.80	5.79	5.78	5.77

Table 11: Sensitivity to the calibration sample count N on LLaMA-2-7B W4A16 (KPQ, WikiText-2 PPL, lower is better). All other hyperparameters are held fixed at the defaults ($C=80$, $\alpha_{\min}=0.80$, $\text{seqlen}=2,048$, $\text{seed}=42$). PPL saturates near $N=128$; the $N=16 \rightarrow 128$ change reduces PPL by 0.11, while $N=128 \rightarrow 512$ improves it by only 0.02. We use $N=128$ throughout the paper as the best calibration cost / quality trade-off.

Calibration sample count. Performance saturates around $N=128$; increasing to 512 yields only 0.02 PPL improvement. We use $N=128$ for all experiments.

C (candidates)	10	20	40	80	160
Wiki2 PPL↓	5.80	5.78	5.77	5.79	5.79

Table 12: Sensitivity to the KPQ grid candidate count C on LLaMA-2-7B W4A16 (WikiText-2 PPL, lower is better). Each C uniformly partitions the search range $\alpha \in [\alpha_{\min}, 1.0] = [0.80, 1.0]$. PPL is essentially flat across $C \in [10, 160]$ (range 5.77–5.80, max spread 0.03), indicating the optimum is well-resolved already at $C = 40$. We use $C = 80$ as default throughout to balance robustness and the cost of additional kernel-MSE evaluations.

Grid granularity. PPL is stable across $C \in [10, 160]$; we use $C = 80$ as default.

J AWQ Incompatibility Analysis

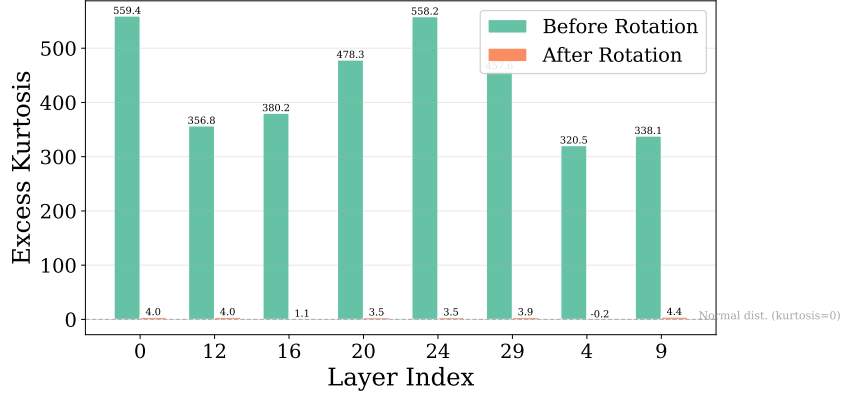


Figure 8: Hadamard rotation collapses per-channel activation kurtosis from a mean of 431 (range 321–559 across 8 representative LLaMA-2-7B layers) to 3.03 (range -0.2 – 4.4), a 99.3% reduction. This eliminates the heavy-tailed salient-channel signal that AWQ uses to identify “important” channels for protective scaling, providing direct evidence that AWQ’s heuristic is no longer informative after model-level rotation.

Before rotation, activation kurtosis averages 431 (range 321–559). After Hadamard rotation, it drops to 3.03 (range -0.2 to 4.4), eliminating the salient-channel signal AWQ uses for protection. Across 8 representative layers, AWQ pre-scaling shifts the KPQ-optimal shrinkage ratio by up to 0.159, confirming a direct conflict between the two scaling strategies.

K Kernel Matrix Visualization

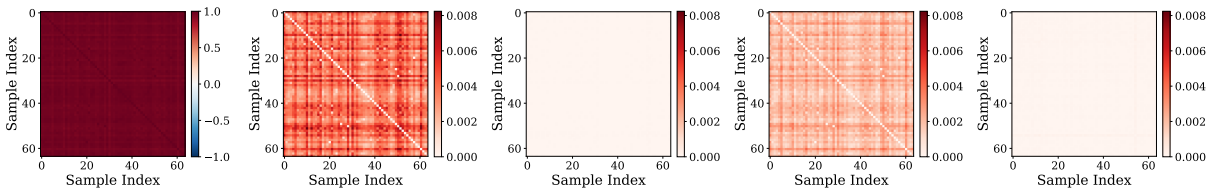


Figure 9: Cosine-similarity kernel matrices $K(\mathbf{Y}) \in \mathbb{R}^{64 \times 64}$ computed at a representative middle layer of LLaMA-2-7B over 64 WikiText-2 calibration samples, under FP16, RTN (W4A16), KPQ-S (W4A16, Step 1 only), and KPQ (W4A16, full). Color encodes pairwise cosine similarity in $[-1, 1]$. RTN visibly degrades the block-diagonal pairwise-similarity structure preserved by FP16; KPQ-S restores most of it through kernel-aware scale selection; the full KPQ is near-indistinguishable from FP16. This visualises the metric that KPQ optimises in Equation 4 and provides direct qualitative support for the kernel-preservation hypothesis.