WARPED-SPACE QUANTIZATION: FROM LAYERWISE ERROR JUMPS TO DISTRIBUTION-MATCHED WEIGHT QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The growing scale of large language models imposes substantial memory and computation costs, making weight quantization an important approach for efficient deployment in practice. Quantization represents continuous weights with a limited set of discrete levels, and its effectiveness increasingly depends on how this limited precision is allocated as the bit width decreases. Existing methods improve this allocation by optimizing rounding, scaling, codebooks, or transforms, but they are commonly designed from static statistics collected from weights or calibration data. During model execution, however, quantization error evolves dynamically: fixed weight residuals interact with changing activations that may already contain perturbations propagated from upstream layers. To characterize this process, we progressively replace full-precision weights with their quantized counterparts and compare the resulting intermediate activations against those of the original model. We observe *jump degradation*, in which activation error rises abruptly at a small number of layers and continues to amplify downstream. Further analysis identifies the mismatch between equally spaced quantization levels and non-uniform weight distributions as one controllable source of the residual error involved in this process. We therefore propose *Warped-Space Quantization* (WSQ), which maps weights into a flatter space for uniform sampling and applies the inverse map to construct distribution-matched non-uniform levels. By allocating more of the limited precision to high-density regions, WSQ reduces the input-independent weight residual before it interacts with changing activations. Experiments across multiple models and benchmarks show that WSQ consistently improves quantized-model quality and outperforms the compared baselines under directly comparable low-bit quantization settings, including extreme regimes.

1 INTRODUCTION

The parameter scale of large language models creates substantial deployment costs across deployment scenarios, making weight quantization one of the most direct compression approaches. Existing methods mainly study how to map weights onto a finite set of levels: GPTQ (Frantar et al., 2023a) and AWQ (Lin et al., 2024) optimize rounding or scaling, NF4 (Dettmers et al., 2023a) and AQLM (Egiazarian et al., 2024) modify the data type or codebook, and QuIP# (Tseng et al., 2024) improves quantization conditions through a transform. Despite these different designs, most quantizers are constructed from statistics collected before deployment or during calibration, including weight distributions, input activations, and Hessian information. These statistics characterize local reconstruction difficulty or layer sensitivity, but provide only a static view of quantization error.

Prior sensitivity and activation-aware studies have shown that the effect of a weight perturbation varies across layers and input states (Shen et al., 2019; Dong et al., 2019b;a; Nagel et al., 2020; Li et al., 2021; Wei et al., 2023; Lin et al., 2024; Xiao et al., 2024; Shao et al., 2024). This dependence becomes important when the quantized model is executed: a fixed weight residual interacts with activations that vary across samples and already contain perturbations propagated from upstream layers. Consequently, the realized quantization error is not a fixed property of an individual layer, but a dynamic process jointly determined by the residual, the current input state, and preceding

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

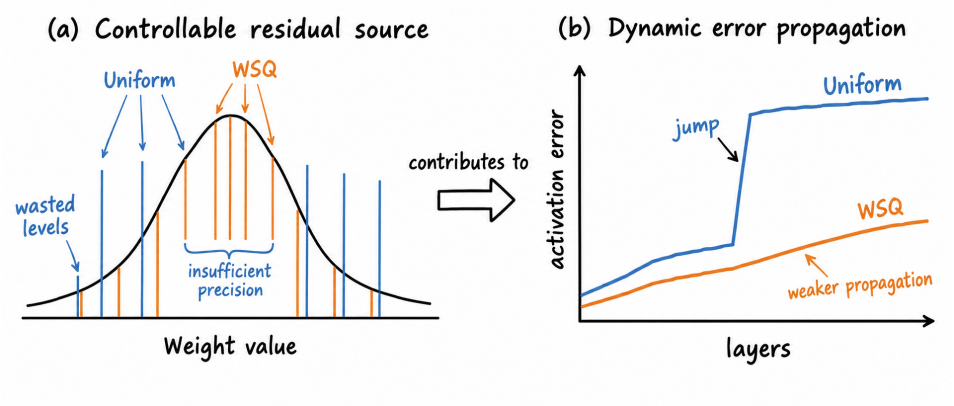


Figure 1: Left: WSQ matches quantization levels to the weight density, reducing the mismatch caused by uniform spacing. Right: the reduced residual is associated with weaker error jumps and downstream amplification.

quantization errors. A calibration-time score can indicate where the model may be sensitive, but cannot directly show how the error evolves as quantized computation proceeds through the network.

To observe this process, we progressively replace full-precision weights with their quantized counterparts and compare the intermediate activations with those of the original model. The error does not always accumulate smoothly: it can rise abruptly at a small number of layers and continue to grow downstream. We call this dynamic pattern *jump degradation*. As illustrated in Figure 1, the measured trajectory contains both an abrupt increase at a layer and continued amplification across subsequent layers. This observation complements static sensitivity analysis: whereas a sensitivity score estimates the response of a layer under a fixed calibration setting, progressive tracking reveals where the realized error changes abruptly and how it propagates through the network.

We next examine which properties of the quantization residual are associated with the observed jumps. Spectral analysis does not support the hypothesis that the measured jump layers can be explained solely by a small number of dominant residual directions. In the cases studied, however, these layers tend to exhibit greater weight-residual energy under uniform quantization, together with center-concentrated or heavy-tailed weight distributions. This evidence suggests that the mismatch between equally spaced levels and non-uniform weight density is one controllable contributor to dynamic error amplification, rather than its sole cause. Uniform levels allocate insufficient resolution to dense regions while spending codewords in sparse tails, which can increase the residual entering the propagation process. Non-uniform quantization does not in itself eliminate jump degradation, because the realized error also depends on the input state, upstream perturbations, and subsequent network transformations. Nevertheless, matching the quantization levels to the weight distribution can reduce the portion of the initial residual caused by an inefficient allocation of finite precision.

Motivated by this finding, we propose **Warped-Space Quantization (WSQ)**. WSQ maps weights into a flatter space for uniform sampling and applies the inverse map to obtain distribution-matched non-uniform levels. By allocating more precision to high-density regions, WSQ reduces the input-independent weight residual before it interacts with changing activations. In the measured trajectories, this reduction is accompanied by weaker error jumps and downstream amplification. Our main contributions are:

- We identify *jump degradation*, a dynamic low-bit quantization pattern in which activation error rises abruptly at a few layers and continues to amplify downstream, and characterize this behavior through progressive layerwise tracking of intermediate activations.
- We identify level–distribution mismatch as a controllable contributor to this process and propose WSQ to construct distribution-matched levels through a warp–sample–inverse-warp procedure.
- We evaluate WSQ across multiple model architectures, bit widths, and datasets, demonstrating consistent improvements over the compared quantization baselines in weight reconstruction, activation fidelity, and overall model quality.

2 Related Work

2.1 Outliers and Distribution-Aware Quantization

Existing methods address non-uniform weight and activation distributions through outlier handling, reconstruction optimization, and alternative quantization spaces. SmoothQuant (Xiao et al., 2024) transfers activation outliers through equivalent channel scaling; AWQ (Lin et al., 2024) and OmniQuant (Shao et al., 2024) adjust scales using activation statistics; and SpQR (Dettmers et al., 2023b) retains a small set of weight outliers at higher precision. OBQ (Frantar et al., 2023b) and GPTQ (Frantar et al., 2023a) optimize rounding and error compensation, while NF4 (Dettmers et al., 2023a), AQLM (Egiazarian et al., 2024), LeanQuant (Zhang & Shrivastava, 2025), and QuIP# (Tseng et al., 2024) improve the data type, codebook, quantization levels, or transformation geometry. These approaches improve weight representation under severe precision constraints using statistics available before execution, but do not directly characterize how the resulting residual evolves with changing activations. WSQ instead begins from the observed layerwise error trajectory and uses warped-space sampling to reduce the distribution-induced component of the residual.

2.2 Quantization Sensitivity and Error Propagation

Q-BERT (Shen et al., 2019), HAWQ (Dong et al., 2019b), and HAWQ-V2 (Dong et al., 2019a) estimate layer sensitivity from Hessian information, whereas AdaRound (Nagel et al., 2020), BRECQ (Li et al., 2021), and QDrop (Wei et al., 2023) optimize rounding through local reconstruction objectives. Together with activation-aware methods, they show that quantization effects vary across layers and input conditions. However, their sensitivity estimates are generally computed during calibration and then used as fixed guidance, so they do not directly reveal how realized error changes after upstream perturbations enter the layer input. We therefore track intermediate activations throughout the quantized network, exposing abrupt error growth and downstream amplification, and use the resulting trajectory to identify the controllable residual component targeted by WSQ.

3 Method

A quantized weight matrix contains a fixed approximation residual, but the activation error induced by this residual changes with the current input and with perturbations propagated from preceding layers. We first characterize this dynamic behavior through layerwise activation trajectories. We then examine which properties of the weight residual are associated with the observed error jumps and isolate the component that can be controlled directly by the quantizer. This analysis leads to WSQ, which reshapes the quantization space so that finite reconstruction capacity can be more effectively allocated according to both the distribution and geometric structure of the weights.

3.1 Jump Degradation

Let $e_{\text{act}}^{(l)} = \text{MSE}(\mathbf{h}_{\text{fp}}^{(l)}, \mathbf{h}_q^{(l)})$ denote the activation error at layer l. We measure its layerwise increment as

$$\Delta e^{(l)} = e_{\text{act}}^{(l)} - e_{\text{act}}^{(l-1)}. \tag{1}$$

A layer is identified as a jump point when $\Delta e^{(l)}$ exceeds the robust IQR threshold $Q_3 + 1.5 \times \text{IQR}$ computed from the empirical increment distribution (Tukey, 1977). This criterion distinguishes abrupt changes from ordinary layer-to-layer fluctuations without assuming normally distributed errors.

We obtain the trajectory by progressively replacing full-precision weights with their quantized counterparts and comparing the resulting intermediate activations with those of the original model. The measured trajectories exhibit two representative patterns: an isolated abrupt increase at a single layer and continued amplification across several adjacent layers. We refer to them as *sudden* and *cascading* patterns, respectively; they are observations of the measured trajectories rather than exhaustive categories. The same criterion can be applied at neuron granularity after a jump layer is located.

Under extreme low-bit quantization, the anomalous neurons identified at the same layer overlap only partially across datasets, and their error magnitudes also change with the input. This result shows

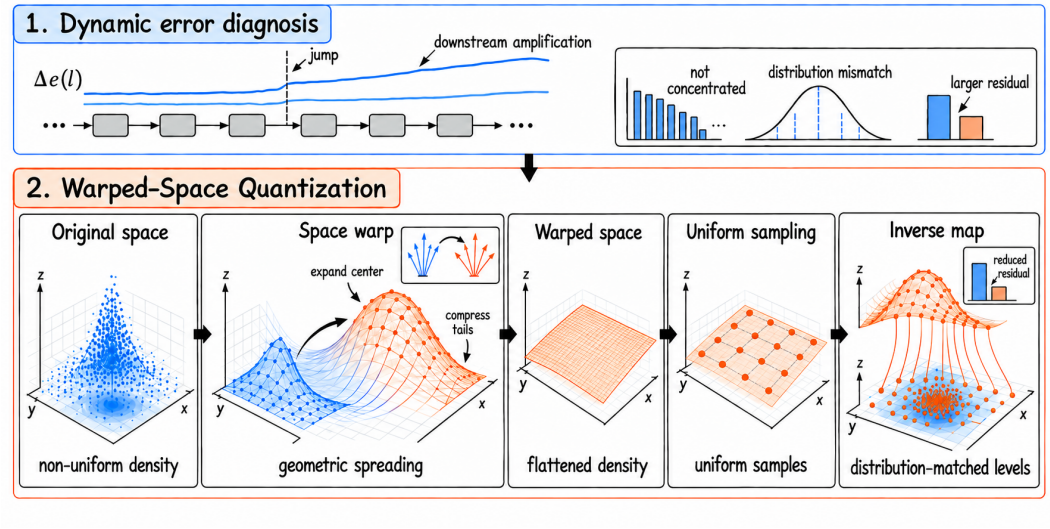


Figure 2: Top: layerwise tracking reveals dynamic error jumps, while residual analysis identifies level–distribution mismatch as a controllable error source. Bottom: WSQ warps the non-uniform weight space into a flattened geometry, performs uniform sampling in the warped space, and maps the samples back to obtain distribution-matched levels with reduced residual error.

that jump degradation cannot be treated as a fixed property of the weight residual alone. The residual is fixed after quantization, whereas the activation deviation it produces depends on the current input state and accumulated upstream perturbations.

3.2 Residual Energy and Dynamic Error Amplification

Let $\mathbf{E}^{(l)} = \mathbf{W}^{(l)} - Q(\mathbf{W}^{(l)})$ denote the weight residual at layer l. For a linear projection, the locally induced activation deviation satisfies

$$\begin{aligned} \delta\mathbf{h}^{(l)} &\approx \mathbf{E}^{(l)}\mathbf{h}^{(l-1)}, \\ \left\|\delta\mathbf{h}^{(l)}\right\|_2^2 &= \left(\mathbf{h}^{(l-1)}\right)^\top \mathbf{G}^{(l)}\mathbf{h}^{(l-1)}, \qquad \mathbf{G}^{(l)} = \left(\mathbf{E}^{(l)}\right)^\top \mathbf{E}^{(l)}. \end{aligned} \tag{2}$$

For layers containing normalization and nonlinear operations, this relation serves as a local first-order approximation. The spectrum of $\mathbf{G}$ describes how residual energy is distributed across input directions. If a jump were dominated by a small residual subspace, a few eigenvalues would account for most of the energy. In our measurements, however, the effective ranks of the examined non-jump, sudden-pattern, and cascading-pattern layers remain close to full rank. Although this does not exclude alignment with particular inputs, it provides no evidence that the observed jumps are explained by a low-rank residual structure.

The total residual energy instead bounds the local deviation:

$$\left\|\delta\mathbf{h}^{(l)}\right\|_2^2 \le \left\|\mathbf{E}^{(l)}\right\|_F^2 \left\|\mathbf{h}^{(l-1)}\right\|_2^2. \tag{3}$$

A large residual norm is neither necessary nor sufficient for downstream cascading, which also depends on the input state and subsequent transformations. Nevertheless, for a fixed input energy, reducing $\|\mathbf{E}^{(l)}\|_F^2$ tightens the worst-case local error bound and is directly controllable by the weight quantizer.

LLM weights are typically concentrated near zero with relatively sparse tails. Uniform spacing therefore provides insufficient resolution where weights are dense while spending levels where weights occur less frequently. Consistent with this explanation, the layerwise tracking results in Table 3 show that distribution-matched quantization reduces mean weight MSE relative to uniform quantization. These observations identify level–distribution mismatch as a controllable source of the residual involved in dynamic error propagation, rather than as the sole cause of jump degradation.

3.3 WARPED-SPACE QUANTIZATION

The preceding analysis identifies weight-residual energy as the component of dynamic error propagation that can be controlled directly by the quantizer. Uniform quantization is inefficient for this objective because LLM weights are not uniformly distributed: most values concentrate near the center, whereas relatively few occupy the tails. Equal spacing therefore assigns the same resolution to regions with very different probability mass. WSQ addresses this mismatch by moving quantization from the original weight space to a warped space in which finite reconstruction capacity can be distributed more effectively.

Distributional warping. Let $p(w)$ denote the weight density and let $c : \mathbb{R} \to \mathbb{R}$ be a monotonic magnitude transform. A scalar weight is first mapped to

$$z = c(w), \tag{4}$$

quantized uniformly in the warped coordinate, and mapped back through

$$\hat{w} = c^{-1}(Q_{\text{uni}}(c(w))) \,. \tag{5}$$

Uniform intervals in the warped space become non-uniform reconstruction intervals in the original space. A larger derivative $c'(w)$ expands a region before sampling and consequently places more reconstruction levels in that region after the inverse map. WSQ therefore uses finer resolution in dense regions and wider intervals in sparse tails.

Under the classical high-resolution approximation for squared-error distortion, the optimal scalar point density is proportional to $p(w)^{1/3}$, which gives

$$c'(w) \propto p(w)^{1/3} \tag{6}$$

(Bennett, 1948; Lee & Song, 2025). This relation provides the distributional principle behind WSQ: the warped coordinate expands regions in proportion to their contribution to expected reconstruction error rather than preserving equal distances in the original weight space.

Geometric warping. Matching the marginal distribution alone does not characterize how neighboring weights are arranged in a high-dimensional space. Correlated coordinates or localized outliers may remain difficult to represent even when each marginal distribution receives an appropriate level density. WSQ therefore also permits an orthogonal geometric transform $\mathbf{H}$, instantiated with blockwise Hadamard rotations and random sign flips (Chee et al., 2024; Tseng et al., 2024). The rotation redistributes localized structure across coordinates while preserving the Euclidean norm, producing a more balanced representation before sampling. Distributional companding and geometric rotation address complementary aspects of the same problem: the former reallocates resolution along the magnitude axis, whereas the latter reshapes how error-prone directions are distributed across coordinates.

Sampling in the warped space. After transformation, WSQ may sample either individual coordinates or d-dimensional weight vectors. For scalar sampling, we construct K reconstruction levels using Lloyd–Max optimization (Lloyd, 1982; Max, 1960), which alternates between

$$b_k = \frac{r_k + r_{k+1}}{2}, \qquad r_k = \mathbb{E}[w \mid b_{k-1} \le w < b_k] \,, \tag{7}$$

to minimize $\mathbb{E}[(w - Q(w))^2]$. Unlike quantile levels, which assign equal probability mass to each interval, Lloyd–Max levels directly minimize the residual-energy objective. We compute a global level table from a standard-Gaussian reference distribution, normalize it to $[-1, 1]$, and share it across weight groups. Each group applies absmax scaling, maps the normalized weights to their nearest levels, and restores the original scale; no learned per-group codebook is required.

For vector sampling, adjacent transformed weights are grouped into d-dimensional vectors and assigned to learned codewords. This allows the sampler to represent joint structure that cannot be captured by independent scalar levels (Zador, 1982; Egiazarian et al., 2024). Scalar and vector sampling are therefore two realizations of the same warped-space principle rather than separate quantization methods.

Unified WSQ operator. Let T denote the invertible composition of magnitude warping and geometric rotation, and let Λ specify the reconstruction set used for scalar or vector sampling. WSQ is written uniformly as

$$\hat{\mathbf{W}} = T^{-1}(Q_\Lambda(T(\mathbf{W}))) \,. \tag{8}$$

The choice of T determines the geometry in which distances are measured, whereas Λ determines how the finite reconstruction budget is allocated in that geometry. A scalar realization uses magnitude companding with a global Lloyd–Max level set; a geometric realization additionally applies Hadamard rotation and may use a learned vector codebook. In both cases, the reconstructed weights are mapped back to the original space before inference, with no input adaptation.

WSQ does not adapt its quantization levels to each runtime input. Instead, it reduces the input-independent weight residual before that residual interacts with changing activations. The same transformed weights can therefore be used across input states, while the reduced residual lowers the controllable error injected into their dynamic propagation.

4 Experiments

4.1 Experimental Setup

Environment. All experiments are conducted on a single NVIDIA A100 80GB GPU.

Models. We evaluate Qwen3 (Yang et al., 2025), Gemma 2 (Gemma Team et al., 2024), and Llama 3.1 (Grattafiori et al., 2024), using a group size of 128 for all models.

Benchmarks. Our evaluation covers language modeling, general language understanding, and agentic tasks. Specifically, we report perplexity (PPL) on WikiText-2 (Merity et al., 2016); performance on MMLU (Hendrycks et al., 2021), MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), AdvGLUE (Wang et al., 2021), and TruthfulQA (Lin et al., 2022); and results on MatPlotAgent (Yang et al., 2024), ChinaTravel (Shao et al., 2025), and SimuHome (Seo et al., 2025).

4.2 Experimental Results

Main results. Table 1 compares WSQ with representative post-training quantization methods on Qwen3-8B. WSQ achieves the best overall performance across the 4-, 3-, and 2-bit settings, with its advantage becoming more pronounced at lower bit widths. At 4 bits, WSQ obtains a WikiText-2 perplexity of 8.353, reducing the best baseline result of 8.925 by 6.4%, while achieving the best results on most language-understanding and agentic benchmarks. At 3 bits, WSQ further reduces perplexity from 9.334 to 8.945 and substantially improves MNLI, QNLI, MatPlotAgent, and SimuHome. At 2 bits, WSQ achieves a perplexity of 10.916, a 15.2% reduction relative to LeanQuant, and obtains the best results on nearly all downstream tasks. Although WSQ is not uniformly superior on TruthfulQA at 3 and 4 bits, its consistently strong performance across the remaining benchmarks demonstrates a better overall trade-off between language modeling, language understanding, and agentic capabilities. These results support our motivation that distribution-matched reconstruction levels are particularly beneficial when the quantization budget is severely limited.

Table 1: Performance comparison of quantization methods on Qwen3-8B across different bit widths. TQA reports MC1/MC2 scores. The best result within each bit-width group is highlighted in bold.

Bits	Method	PPL↓	MMLU↑	MNLI↑	QNLI↑	AdvGLUE↑	TQA↑	MatPlot↑	SimuHome↑	ChinaTr.↑
16	**FP16**	**7.801**	**0.748**	**0.805**	**0.913**	**0.716**	**0.351/0.552**	**0.363**	**0.425**	**0.011**
4	AWQ	8.925	0.735	0.786	0.839	0.698	**0.351/0.553**	0.304	0.388	0.006
	OmniQuant	9.358	0.725	0.801	0.556	0.695	0.342/0.548	0.290	0.343	0.003
	LeanQuant	9.000	0.736	0.776	0.896	0.706	0.338/0.538	0.267	**0.425**	0.000
	WSQ (Ours)	**8.353**	**0.748**	**0.802**	**0.897**	**0.709**	0.338/0.538	**0.321**	**0.425**	**0.009**
3	OmniQuant	11.394	0.639	0.644	0.540	0.589	0.322/0.524	0.200	0.317	0.002
	LeanQuant	9.334	0.692	0.711	0.840	0.628	**0.361/0.554**	0.050	0.150	0.000
	WSQ (Ours)	**8.945**	**0.696**	**0.785**	**0.872**	**0.633**	0.294/0.471	**0.229**	**0.372**	**0.006**
2	OmniQuant	124.47	0.231	0.358	0.506	0.432	0.269/0.450	0.110	0.000	0.000
	LeanQuant	12.880	0.441	0.543	0.512	0.476	0.296/0.481	0.000	0.000	0.000
	WSQ (Ours)	**10.916**	**0.509**	**0.544**	**0.607**	**0.480**	**0.301/0.533**	**0.133**	0.000	**0.026**

Cross-model results. Table 2 evaluates the adapted WSQ pipeline on Qwen3-14B, Gemma 2 9B, and Llama 3.1 8B. At 3 bits, WSQ achieves perplexities of 7.230, 9.358, and 7.303 on the three

Model	Method	bits	PPL↓	MMLU↑	MNLI↑	QNLI↑	adv_glue↑	TQA↑
Qwen3-14B	FP16	16	8.128	0.540	0.820	0.840	0.780	0.320/0.488
	AWQ	4	8.063	0.480	0.800	0.860	0.780	0.300/0.462
	WSQ	3	**7.230**	0.460	0.720	0.780	0.800	0.280/0.519
	WSQ	2	**8.970**	0.400	0.560	0.580	0.700	0.220/0.390
Gemma 2 9B	FP16	16	8.910	0.560	0.820	0.880	0.740	0.420/0.612
	AWQ	4	9.330	0.540	0.820	0.860	0.760	0.380/0.589
	WSQ	3	**9.358**	0.480	0.820	0.860	0.760	0.320/0.601
	WSQ	2	**11.510**	0.440	0.680	0.680	0.740	0.320/0.509
Llama 3.1 8B	FP16	16	7.244	0.400	0.700	0.740	0.840	0.280/0.503
	AWQ	4	7.513	0.360	0.720	0.660	0.800	0.280/0.494
	WSQ	3	**7.303**	0.220	0.640	0.500	0.760	0.240/0.436
	WSQ	2	**11.274**	0.320	0.520	0.480	0.360	0.280/0.471

Table 2: Cross-architecture adapted-pipeline results. PPL uses WikiText-2 test, while the other tasks use a 50-sample exploratory evaluation.

models, respectively, remaining comparable to or outperforming the corresponding 4-bit AWQ results despite using one fewer bit. At 2 bits, WSQ maintains finite perplexities of 8.970, 11.510, and 11.274, although the downstream scores exhibit greater task- and architecture-dependent variation under such aggressive compression. Since the non-perplexity metrics are based on a 50-sample exploratory evaluation, they are primarily intended to indicate cross-model trends rather than provide precise benchmark estimates. Overall, the results show that WSQ generalizes beyond Qwen3-8B to different model scales and architectural families, while retaining its effectiveness under extreme low-bit quantization.

4.3 Jump Degradation and Layerwise Error Propagation

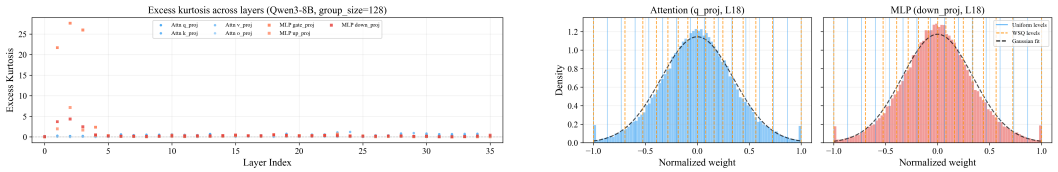
This experiment examines how quantization error evolves across network depth and whether distribution-matched quantization changes the resulting propagation trajectory. We apply uniform quantization and WSQ to all 36 decoder layers of Qwen3-8B under matched 4-bit weight-only settings, using the same group size, target modules, and evaluation inputs. At each layer, we compare the intermediate activations with those of the full-precision model and record activation MSE, cosine similarity, and the layerwise MSE increment. Jump points are identified by applying the IQR criterion to the empirical increment distribution.

The measured trajectories reveal two characteristic forms of jump degradation. A sudden pattern appears as an isolated MSE spike around L16–17, including a $495\times$ layerwise increase at L16, while a cascading pattern appears as continued 1.3–$1.4\times$ amplification across later adjacent layers. The IQR criterion identifies six jump layers under uniform quantization (L16, L17, L31–L33, and L35) and five under WSQ (L31–L35). Thus, the isolated early jump is absent from the WSQ trajectory, and the remaining jumps are concentrated in the final layers. Table 3 summarizes the overall trajectory. Under WSQ, the early jump region at L16–17 disappears, and the mean activation MSE decreases substantially from 65.2 to 1.60. WSQ also reduces the final-layer activation MSE from 29.1 to 19.5 and the mean weight MSE from 1.12×10^{-5} to 6.95×10^{-6}, while increasing the final-layer cosine similarity from 0.9771 to 0.9839. These results show that WSQ improves the error trajectory beyond an isolated jump point, reducing both weight reconstruction error and activation deviation across the network.

Metric	Uniform	WSQ
Jump locations	L16–17, L31–33, L35	L31–35
Final-layer act. MSE	2.91e+01	1.95e+01
Mean act. MSE	6.52e+01	1.60e+00
Final-layer cos. sim.	0.9771	0.9839
Mean weight MSE	1.12e-05	6.95e-06

Table 3: Layerwise tracking results on Qwen3-8B under 4-bit weight-only quantization with group size 128.

To examine the distributional structure associated with these trajectories, we measure weight kurtosis across layers and compare representative module-wise distributions. Figure 3(a) shows that



(a) Excess kurtosis across layers. (b) Representative module-wise weight distributions.

Figure 3: Distributional characteristics of Qwen3-8B weights. (a) MLP projections, particularly `down_proj`, exhibit higher excess kurtosis than attention projections. (b) Their sharper centers and heavier tails make uniformly spaced quantization levels less efficient in high-density regions.

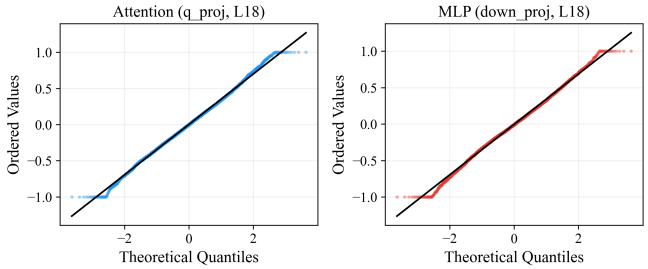


Figure 4: Normal QQ plots of representative weight distributions. Attention projections closely follow the Gaussian reference, whereas MLP `down_proj` deviates in the tails.

MLP projections, particularly `down_proj`, consistently exhibit higher excess kurtosis than attention projections. As shown in Figure 3(b), these modules have sharper centers and heavier tails, so equally spaced levels allocate insufficient resolution to the high-density region. The joint reduction in weight MSE and activation error shows that redistributing quantization levels according to weight density effectively reduces the residual entering the layerwise propagation process.

4.4 Distributional Evidence and the MSE–PPL Gap

We examine whether the Gaussian reference used by WSQ matches the weight distributions of different modules. Shapiro–Wilk tests (Shapiro & Wilk, 1965) on 200 weight groups from Qwen3-14B show that 92%–95% of attention-layer groups pass the normality test, whereas the pass rate of MLP `down_proj` is 0%. The latter also exhibits an excess kurtosis of 3–5, indicating a sharper center and heavier tails. The QQ plots in Figure 4 show the same pattern: attention weights closely follow the reference diagonal, while MLP weights deviate systematically in the tails. These results support Gaussian-derived levels for attention projections and motivate geometric warping for modules with stronger heavy-tailed structure.

We next compare how different level designs affect weight reconstruction and model quality on Qwen3-14B layers 0–9. As shown in Table 4, Lloyd–Max WSQ achieves the lowest weight MSE and PPL among the compared methods. Its weight MSE is 62.5% of uniform quantization, while its PPL decreases from 8.239 to 8.133. Asymmetric quantization also improves both metrics, but requires a slightly higher effective bit rate.

Method	MSE vs. Uniform	PPL	Bits
WSQ (Lloyd–Max)	**62.5%**	**8.133**	4.125
Asymmetric	71.2%	8.138	4.156
μ-law	353.6%	8.154	4.125
Uniform	100.0%	8.239	4.125
WSQ-Quantile	63.2%	8.387	4.125

Table 4: Comparison of weight reconstruction error and perplexity across quantization-level designs on Qwen3-14B. MSE is reported relative to uniform quantization.

The remaining methods reveal that weight MSE and PPL do not always induce the same ordering. Although μ-law companding (ITU-T, 1988) produces the largest weight MSE, its PPL remains lower than that of uniform quantization because it allocates more levels near zero while allowing larger

tail errors. Conversely, WSQ-Quantile obtains an MSE close to Lloyd–Max WSQ but produces the highest PPL in the comparison. Quantile levels match cumulative probability mass, whereas Lloyd–Max centroids directly minimize pointwise squared reconstruction error. These results support residual energy as a directly controllable reconstruction objective while showing that end-to-end quality also depends on how the residual is distributed across weights and network computations.

4.5 Interaction with Rounding Optimization

We examine whether WSQ and GPTQ (Frantar et al., 2023a) provide complementary gains.

In a 2×2 comparison of $\{\text{RTN}, \text{GPTQ}\} \times \{\text{Uniform}, \text{WSQ}\}$, WSQ reduces PPL by 0.106 under RTN but by only 0.016 under GPTQ (Figure 5). Meanwhile, GPTQ improves Uniform by 0.079 but slightly degrades WSQ by 0.010. Thus, their gains do not combine additively.

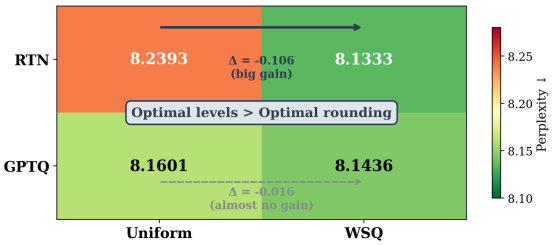


Figure 5: Interaction between rounding optimization and quantization-level design. WSQ provides a smaller gain after GPTQ compensation.

A possible explanation is that GPTQ's rounding updates move the reconstructed weights away from the distribution used to derive the Lloyd–Max levels (Lloyd, 1982; Max, 1960). This observation suggests that rounding optimization should be adapted to distribution-matched levels, although repeated runs are needed to establish the interaction.

4.6 Efficiency Analysis

Under a shared 4-bit, group-size-128 setting, Figure 6 compares deployment storage and quantization-time memory at 8B and 14B scales. WSQ has the same checkpoint size as RTN—4.13 GB and 7.22 GB, respectively—because its quantization levels are shared globally. It avoids both the per-matrix codebook overhead of AQLM and the calibration or second-order workspace required by AWQ, GPTQ, and AQLM. These results show that WSQ scales with model size without introducing method-specific storage or calibration-memory overhead.

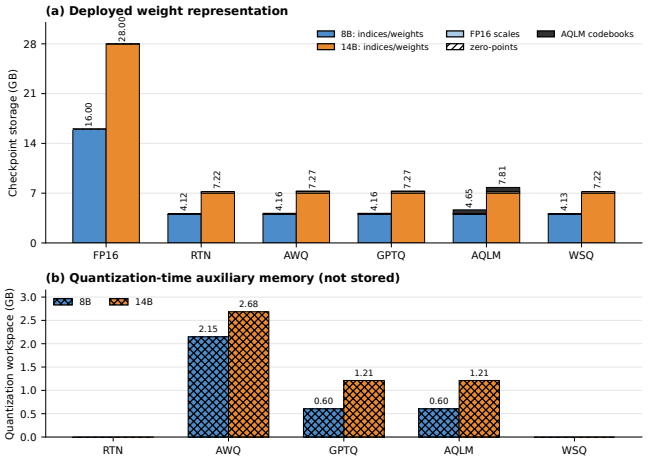


Figure 6: All methods use 4-bit weights with $G = 128$. The lower panel reports temporary buffers excluded from checkpoint size.

5 Conclusion

We identify *jump degradation*, where quantization-induced activation error rises abruptly at a few layers and is amplified downstream. Our analysis suggests that the mismatch between uniformly spaced levels and non-uniform weight distributions is a controllable contributor to this process, rather than its sole cause. Based on this insight, we propose Warped-Space Quantization (WSQ), which constructs distribution-matched levels to reduce weight residuals before they interact with changing activations. Experiments across multiple models, bit widths, and benchmarks show that WSQ consistently reduces reconstruction and activation errors while improving overall model quality, with particularly clear gains under extreme low-bit quantization.

REFERENCES

W. R. Bennett. Spectra of quantized signals. *Bell System Technical Journal*, 27(3):446–472, 1948.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. QuIP: 2-bit quantization of large language models with guarantees, 2024. URL https://arxiv.org/abs/2307.13304.

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

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs, 2023a. URL https://arxiv.org/abs/2305.14314.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. SpQR: A sparse-quantized representation for near-lossless LLM weight compression, 2023b. URL https://arxiv.org/abs/2306.03078.

Zhen Dong, Zhewei Yao, Yaohui Cai, Daiyaan Arfeen, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ-V2: Hessian aware trace-weighted quantization of neural networks, 2019a. URL https://arxiv.org/abs/1911.03852.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael Mahoney, and Kurt Keutzer. HAWQ: Hessian AWare quantization of neural networks with mixed-precision, 2019b. URL https://arxiv.org/abs/1905.03696.

Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization, 2024. URL https://arxiv.org/abs/2401.06118.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers, 2023a. URL https://arxiv.org/abs/2210.17323.

Elias Frantar, Sidak Pal Singh, and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning, 2023b. URL https://arxiv.org/abs/2208.11580.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, et al. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

ITU-T. Recommendation G.711: Pulse code modulation (PCM) of voice frequencies. International Telecommunication Union, 1988.

Deokjae Lee and Hyun Oh Song. Q-palette: Fractional-bit quantizers toward optimal bit allocation for efficient llm deployment, 2025. URL https://arxiv.org/abs/2509.20214.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the limit of post-training quantization by block reconstruction, 2021. URL https://arxiv.org/abs/2102.05426.

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 LLM compression and acceleration, 2024. URL https://arxiv.org/abs/2306.00978.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods, 2022. URL https://arxiv.org/abs/2109.07958.

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, 2016. URL https://arxiv.org/abs/1609.07843.

Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization, 2020. URL https://arxiv.org/abs/2004.10568.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text, 2016. URL https://arxiv.org/abs/1606.05250.

Gyuhyeon Seo, Jungwoo Yang, Junseong Pyo, Nalim Kim, Jonggeun Lee, and Yohan Jo. SimuHome: A temporal- and environment-aware benchmark for smart home LLM agents. *CoRR*, abs/2509.24282, 2025. URL https://doi.org/10.48550/arXiv.2509.24282.

Jie-Jing Shao, Bo-Wen Zhang, Xiao-Wen Yang, Baizhi Chen, Siyu Han, Wen-Da Wei, Guohao Cai, Zhenhua Dong, Lan-Zhe Guo, and Yu-Feng Li. ChinaTravel: An open-ended benchmark for language agents in chinese travel planning. In *Workshop on Scaling Environments for Agents*, 2025. URL https://openreview.net/forum?id=dPMlVo3rNy.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. OmniQuant: Omnidirectionally calibrated quantization for large language models, 2024. URL https://arxiv.org/abs/2308.13137.

S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. Q-BERT: Hessian based ultra low precision quantization of BERT, 2019. URL https://arxiv.org/abs/1909.05840.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: Even better LLM quantization with Hadamard incoherence and lattice codebooks, 2024. URL https://arxiv.org/abs/2402.04396.

John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL https://openreview.net/forum?id=GF9cSKI3A_q.

Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. QDrop: Randomly dropping quantization for extremely low-bit post-training quantization, 2023. URL https://arxiv.org/abs/2203.05740.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference, 2018. URL https://arxiv.org/abs/1704.05426.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhiyu Yang, Zihan Zhou, Shuo Wang, Xin Cong, Xu Han, Yukun Yan, Zhenghao Liu, Zhixing Tan, Pengyuan Liu, Dong Yu, Zhiyuan Liu, Xiaodong Shi, and Maosong Sun. MatPlotAgent: Method and evaluation for LLM-based agentic scientific data visualization. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. URL https://aclanthology.org/2024.findings-acl.701/.

Paul L. Zador. Asymptotic quantization error of continuous signals and the quantization dimension. *IEEE Transactions on Information Theory*, 28(2):139–149, 1982.

Tianyi Zhang and Anshumali Shrivastava. LeanQuant: Accurate and scalable large language model quantization with loss-error-aware grid. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ISqx8giekS.

A JUMP DETECTION HYPERPARAMETER SENSITIVITY

Experimental setup. For Qwen3-14B L20 and L25 q_proj, we quantize to 2-bit then scan per-neuron cosine similarity drop (gradient), obtaining the error gradient distribution over 500 neurons. The gradient distribution is severely non-normal—L20 kurtosis 117, L25 kurtosis 124 (a normal distribution has 3)—indicating that the 3σ method is inapplicable. We compare three robust threshold methods under different parameters: IQR ($Q_3 + k \times \text{IQR}$, $k \in \{1.0, 1.5, 2.0, 2.5, 3.0\}$), 3σ ($\mu + k \times \sigma$, $k \in \{2, 3, 4\}$), and MAD (med $+ k \times$ MAD $\times 1.4826$, $k \in \{2, 3, 4\}$).

Detection results.

Table 5: Jump detection hyperparameter sensitivity

Threshold	L20 jumps	L20 %	L25 jumps	L25 %	L20 avg sev.
IQR(1.0)	59	11.8%	35	7.0%	5.49
IQR(1.5)	**39**	**7.8%**	**24**	**4.8%**	**5.79**
IQR(2.0)	29	5.8%	19	3.8%	5.93
IQR(2.5)	26	5.2%	17	3.4%	5.42
IQR(3.0)	20	4.0%	12	2.4%	5.76
3σ(2)	4	0.8%	3	0.6%	3.18
3σ(3)	2	0.4%	3	0.6%	3.44
3σ(4)	2	0.4%	3	0.6%	2.58
MAD(2)	59	11.8%	36	7.2%	5.49
MAD(3)	33	6.6%	22	4.4%	5.90
MAD(4)	26	5.2%	17	3.4%	5.40

Analysis. The 3σ family flags only 0.4%–0.8% of neurons, substantially fewer than the robust rules, because extreme values inflate the standard deviation and move the threshold farther into the tail. IQR and MAD produce similar sets at comparable detection rates: the Jaccard agreement between IQR(1.5) and MAD(3) is 0.85 on L20 and 0.92 on L25. The IQR sweep moves smoothly from a more inclusive threshold at $k = 1.0$ to increasingly conservative thresholds for $k \geq 2.0$.

No independent ground truth for jump neurons is available, so these results do not establish IQR(1.5) as precision–recall optimal. They show instead that IQR(1.5) provides a stable, moderately selective operating point on the two measured layers; we therefore use it as the default empirical threshold.

B CROSS-MODEL AND TASK-LEVEL EVALUATION

The main text reports the cross-model general benchmarks. Here we examine how capability changes with task difficulty. Figure 7 shows the FP16 SimuHome profiles of Qwen3-14B and Llama 3.1 8B, establishing the architecture-dependent task floor used to interpret the method-wise results below.

B.1 DIFFICULTY-STRATIFIED SIMUHOME EVALUATION

Complementing the baseline visualization above, we report method-wise difficulty-stratified SimuHome results on Gemma 2 9B (Gemma Team et al., 2024). All models use the same offline evaluation mode (static action matching) with 20 samples per difficulty level.

Table 6: SimuHome difficulty-stratified scores (Gemma 2 9B, n=20 per level).

Method	bits	L1↑	L2↑	L3↑	L4↑	Overall↑
FP16	16	0.30	0.00	0.15	0.00	0.113
AWQ	4	0.50	0.00	0.00	0.00	0.125
WSQ	3	0.30	0.05	0.00	0.00	0.088
WSQ	2	**0.75**	0.00	0.00	0.00	**0.188**

Key observations. On Gemma 2 9B, 2-bit WSQ obtains the highest L1 and overall scores, 0.75 and 0.188, compared with 0.30/0.113 for FP16 and 0.50/0.125 for 4-bit AWQ. The separation among methods is concentrated at L1: 3-bit WSQ matches FP16, whereas 2-bit WSQ reaches 0.75. Scores

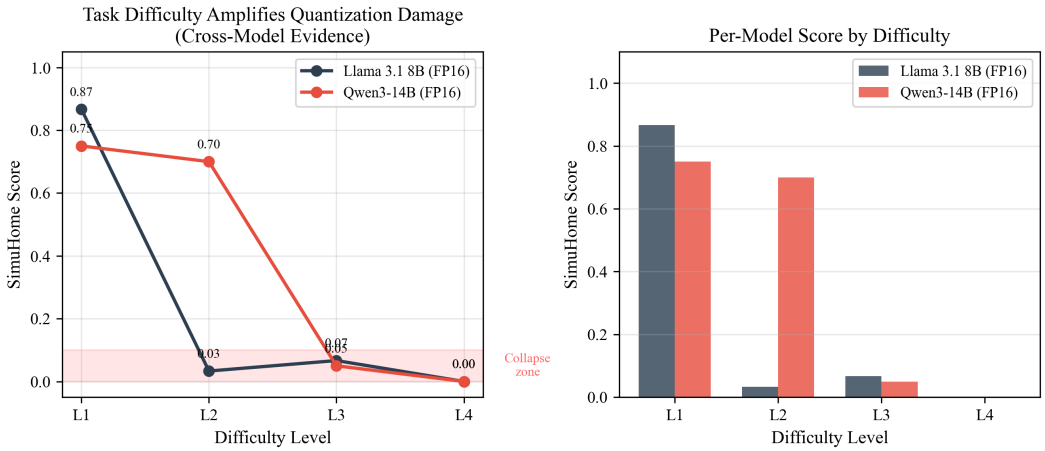


Figure 7: FP16 SimuHome scores across difficulty levels for Qwen3-14B and Llama 3.1 8B, showing architecture-dependent task floors.

approach zero for all methods at L2–L4, including FP16, revealing a task floor at the harder levels rather than a degradation unique to quantized models.

Llama 3.1 8B. Table 7 extends the cross-architecture analysis to Llama 3.1 8B (Grattafiori et al., 2024) (RoPE + GQA), the third architecture in the main text's cross-model comparison.

Table 7: SimuHome difficulty-stratified scores (Llama 3.1 8B, n=30 per level).

Method	bits	L1↑	L2↑	L3↑	L4↑	Overall↑
FP16	16	0.867	0.033	0.067	0.00	0.242
AWQ	4	0.900	0.067	0.167	0.00	0.283
WSQ	3	0.900	0.000	0.033	0.00	0.233
WSQ	2	**0.933**	0.000	0.133	0.00	**0.267**

Key observations. On Llama 3.1 8B, 2-bit WSQ reaches 0.933 at L1, while 3-bit WSQ matches 4-bit AWQ at 0.900. FP16 itself drops from 0.867 at L1 to 0.033 at L2, locating the dominant difficulty cliff in the model–task pair. Above this floor, AWQ retains more capability at L2–L3, whereas WSQ is strongest at L1. Across the evaluated architectures, WSQ consistently preserves the easy-task capability, while the harder levels expose architecture-dependent task floors and method-specific losses.

B.2 Additional Baseline Comparisons

B.2.1 General Benchmarks on Llama 3.1 8B and Gemma 2 9B

We extend the cross-model comparison with OmniQuant (Shao et al., 2024) and QuIP# (Tseng et al., 2024) on Gemma 2 9B and Llama 3.1 8B. PPL is evaluated on the full WikiText-2 test split. MMLU, MNLI, QNLI, AdvGLUE, and TruthfulQA use the same 50-sample protocol to compare capability-retention trends. The 4-bit WSQ rows use scalar PTQ; the 3/2-bit rows use the learned-vector, LoRA-QAT configuration.

Key observations. Scalar WSQ gives the lowest 4-bit PPL on both architectures: 6.93 on Llama 3.1 8B and 8.92 on Gemma 2 9B. The learned-vector WSQ configurations remain stable at 3 and 2 bits, reaching PPL 11.27 and 11.51 at 2 bits. Among the PTQ baselines, QuIP# is substantially more robust than OmniQuant at 2 bits: on Llama 3.1 8B, their PPL values are 12.89 and 5867.43, respectively. This result highlights the importance of Hadamard incoherence and high-dimensional codebooks under extreme compression. The 50-sample task scores complement PPL by showing capability-retention trends under a shared evaluation protocol.

Table 8: Cross-architecture general evaluation (Llama 3.1 8B). PPL↓ on WikiText-2; MMLU/MNLI/QNLI/AdvGLUE/TQA on 50-sample fast eval.

Method	bits	PPL↓	MMLU↑	MNLI↑	QNLI↑	AdvGLUE↑	TQA-MC1↑	TQA-MC2↑
FP16	16	7.244	0.400	0.700	0.740	0.840	0.280	0.503
AWQ	4	7.51	0.360	0.720	**0.660**	0.800	**0.280**	0.494
OmniQuant	4	7.95	0.300	0.680	0.480	0.700	**0.280**	**0.500**
QuIP#	4	7.51	0.400	0.670	0.420	0.800	0.240	0.479
WSQ (ours)	4	**6.93**	**0.460**	**0.740**	**0.660**	**0.840**	**0.280**	0.485
OmniQuant	3	11.12	**0.280**	**0.640**	0.520	0.720	0.200	0.403
QuIP#	3	8.10	0.270	0.580	0.500	0.660	0.220	0.423
WSQ	3	**7.30**	**0.280**	**0.640**	**0.660**	**0.760**	**0.240**	**0.436**
OmniQuant	2	5867.43	0.200	0.280	0.420	0.320	0.220	0.404
QuIP#	2	12.89	0.300	0.300	0.460	0.360	0.180	0.409
WSQ	2	**11.27**	**0.320**	**0.520**	**0.480**	**0.420**	**0.280**	**0.471**

Table 9: Cross-architecture general evaluation (Gemma 2 9B). Same setup as Table 8.

Method	bits	PPL↓	MMLU↑	MNLI↑	QNLI↑	AdvGLUE↑	TQA-MC1↑	TQA-MC2↑
FP16	16	8.91	0.560	0.820	0.880	0.740	0.420	0.612
AWQ	4	9.33	0.540	0.820	**0.860**	0.760	0.380	0.589
OmniQuant	4	14.45	0.540	0.800	0.820	**0.820**	0.420	**0.631**
QuIP#	4	10.99	0.540	0.840	**0.860**	0.740	0.460	0.576
WSQ (ours)	4	**8.92**	**0.560**	**0.880**	**0.860**	0.780	**0.480**	0.598
OmniQuant	3	17.98	0.420	0.800	0.800	0.700	0.380	0.605
QuIP#	3	11.33	0.500	**0.860**	**0.860**	0.720	0.360	0.551
WSQ	3	**9.36**	**0.580**	0.820	**0.860**	**0.760**	**0.420**	**0.606**
OmniQuant	2	97.34	0.200	0.280	0.480	0.520	0.260	0.467
QuIP#	2	21.37	0.260	0.580	0.480	0.600	0.300	0.485
WSQ	2	**11.51**	**0.440**	**0.680**	**0.680**	**0.740**	**0.320**	**0.509**

B.2.2 Edge Task Comparison on Qwen3-14B

We compare WSQ, AWQ, OmniQuant, and QuIP# on Qwen3-14B edge tasks. SimuHome measures instruction following across difficulty levels, while MatPlotAgent and ChinaTravel evaluate structured code generation and commonsense reasoning. Each SimuHome level contains 20 examples; the reported confidence intervals characterize uncertainty at this evaluation scale.

SimuHome difficulty stratification. Table 10 presents per-level SimuHome scores for all quantization methods on Qwen3-14B, with 20 samples per difficulty level (L1–L4) and 95% bootstrap confidence intervals.

Table 10: SimuHome difficulty-stratified scores (Qwen3-14B, n=20 per level). L4 omitted because FP16 scores 0.

Method	bits	L1↑	L2↑	L3↑	Overall↑
FP16	16	0.75	0.70	0.05	0.375
AWQ	4	0.75	**0.85**	0.05	0.413
OmniQuant	4	**0.85**	0.40	**0.15**	0.350
OmniQuant	3	0.70	0.40	0.05	0.288
OmniQuant	2	0.00	0.00	0.00	0.000
QuIP#	3	0.60	0.15	0.00	0.188
QuIP#	2	0.00	0.00	0.00	0.000
WSQ	3	**0.85**	**0.35**	0.00	**0.300**
WSQ	2	**0.55**	**0.05**	0.00	**0.150**

Key observations. WSQ@3 ties for the highest L1 score at 0.85, while AWQ@4 gives the strongest L2 result at 0.85. At 2 bits, WSQ retains scores of 0.55 and 0.05 on L1 and L2, whereas the listed QuIP# and OmniQuant runs score zero. Relative to the stable FP16 transition from L1 to L2 (0.75 to 0.70), OmniQuant@4 and WSQ@3 show larger task-specific drops, from 0.85 to 0.40 and from

0.85 to 0.35. The stratified results therefore distinguish easy-task retention from the additional loss that emerges as task difficulty increases.

Aggregate edge task scores. Table 11 supplements with MatPlotAgent and ChinaTravel scores for WSQ and QuIP#.

Table 11: Aggregate edge task scores (Qwen3-14B). SimuHome: score; MatPlot: plot_completeness; ChinaTravel: all_pass.

Method	bits	SimuHome↑	MatPlot↑	ChinaTr.↑
QuIP#	3	0.188	**0.404**	0.000
QuIP#	2	0.000	0.037	0.000
WSQ	3	**0.300**	0.287	0.000
WSQ	2	**0.150**	**0.033**	0.000

On MatPlotAgent, QuIP# 3-bit achieves higher plot completeness than WSQ (0.404 vs. 0.287), whereas WSQ leads on SimuHome at both 3 and 2 bits. Both methods score zero on ChinaTravel. The results reveal task-dependent strengths: QuIP# better preserves structured code generation in this evaluation, while WSQ better preserves SimuHome instruction following.

C Alternative Error-Reduction Strategies

The main text treats $\|\mathbf{E}\|_F^2$ as a directly controllable error proxy. This appendix examines whether four additional element-level strategies can reduce that proxy under a 4-bit budget.

C.1 Fourier frequency analysis

We apply per-row FFT to the 4-bit quantization residual $\mathbf{R} = \mathbf{W}_{\text{fp16}} - \hat{\mathbf{W}}_{\text{4bit}}$, testing whether residual energy concentrates at low frequencies—if so, a small number of Fourier coefficients could recover most of the error, enabling continuous-basis compression.

Table 12: Fourier residual energy analysis (RTN (Dettmers et al., 2022) 4-bit, group_size=128)

K (freq. components)	Low-freq. energy	MSE recovery	Extra bits/param
4	0.19%	0.17%	+0.025
8	0.35%	0.33%	+0.050
16	0.66%	0.64%	+0.100
64	2.54%	2.52%	+0.400

Residual energy is uniformly distributed across all frequencies; $K = 64$ recovers only 2.52% MSE, exhibiting white-noise characteristics. This suggests that quantization residuals possess little exploitable low-frequency structure, making continuous-basis compression unattractive here.

C.2 Sparse outlier correction

We remove the top-$p\%$ outlier elements by absolute value and store them as FP16, then re-quantize the remaining elements with recomputed group scales. Outlier removal shrinks the group scale, granting higher precision to normal elements (a "quantization sharpening" effect).

Table 13: Sparse outlier correction analysis (RTN 4-bit, group_size=128)

Outlier ratio	Total MSE↓	Scale reduction	Extra bits	Total bits
$p = 0.1\%$	11.1%	2–3%	+0.049	4.174
$p = 0.5\%$	25.6%	7–9%	+0.245	4.370
$p = 1.0\%$	35.2%	11–14%	+0.488	4.613

At $p = 0.1\%$, MSE is reduced by 11.1% at a cost of +0.049 nominal bits per parameter. However, sparse outlier storage introduces values, indices, and irregular memory access. Scalar WSQ instead changes the shared level spacing without adding a per-group sparse correction structure; packed-kernel efficiency is not evaluated by this fake-quant experiment.

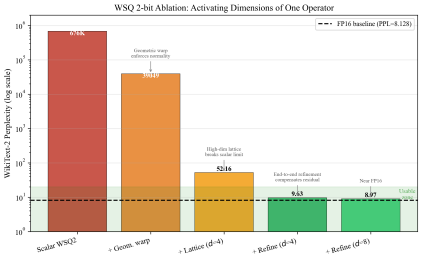


Figure 8: WSQ 2-bit ablation ladder. X-axis: progressive activation of the warped-space operator's dimensions; Y-axis: PPL (log scale). Scalar WSQ2 PPL is extremely high (labeled "collapse"); activating the geometric warp lowers PPL but it remains unusable; grouped d=4 warped-space quantization enters the usable degradation zone; the full d=8 warped-space configuration brings PPL close to the FP16 baseline.

C.3 Residual quantization

We stack k-bit residual quantization on top of 4-bit base quantization ($\mathbf{W} \approx \hat{\mathbf{W}}_{4\text{bit}} + \hat{\mathbf{R}}_{k\text{bit}}$), testing whether residual quantization can significantly reduce error at low additional bit cost. The 4+2-bit scheme reduces MSE by approximately 75%, but the total bit budget reaches 6-bit, exceeding the 4-bit target; 1-bit residual introduces additional error. Residual quantization is suitable for 6-bit and above scenarios, not for 4-bit extreme compression.

C.4 Row pairing transform

We identify row pairs with high cosine similarity in the weight matrix, apply sum/diff orthogonal transforms, then quantize—testing whether the reduced scale of diff rows lowers total MSE. Experiments show that sum/diff transforms do reduce diff-row scales by 40–60% on average, but sum-row scales increase correspondingly—the orthogonal transform only redistributes error energy without reducing the total. This is consistent with the theoretical bound in the main text: orthogonal transforms preserve the Frobenius norm ($\|\mathbf{E}\|_F^2 = \|\mathbf{E}\mathbf{H}^\top\|_F^2$), and cannot reduce the total error energy.

D Low-Bit WSQ Ablation and Implementation Details

Table 14 separates the warped-space configurations used in the nominal 2-bit pipeline. Scalar WSQ2 and a one-dimensional geometric warp alone collapse. Extending the distortion model to grouped d-dimensional warped spaces provides the first large recovery, while the full d=8 warped-space configuration produces the lowest reported PPL. These variants differ only in the geometry in which quantization is performed.

Table 14: WSQ incremental ablation (2-bit)

Configuration	Representation note	PPL↓	MMLU↑	MNLI↑	QNLI↑	adv_glue↑	TQA↑
Scalar WSQ2	fake quant	676159	0.229	0.346	0.494	0.424	0.230
+ Geom. warp	fake quant	39049	0.245	0.332	0.482	0.432	0.214
+ Grouped warp (d=4)	fake quant	52.16	0.340	0.335	0.507	0.442	0.288
+ Full WSQ (d=4)	fake quant	9.630	0.565	**0.582**	0.524	0.544	**0.359**
+ Full WSQ (d=8)	fake quant	**8.970**	**0.646**	0.558	**0.573**	**0.600**	0.319

The ablation identifies a clear transition from scalar to high-dimensional warped-space quantization. Scalar WSQ and the one-dimensional geometric warp collapse at 2 bits; grouped d=4 quantization restores a usable PPL, and the full d=8 configuration reaches 8.970. The results establish that high-dimensional geometry is necessary for the reported 2-bit quality. All rows use fake quantization, so the table evaluates reconstructed-model quality rather than packed-kernel performance.

AI USE DISCLOSURE

Generative AI tools were used solely to improve the clarity, grammar, and readability of the manuscript and to assist with software development, including code completion, debugging, and refactoring. They were not used to formulate the research questions, develop the methodology, derive scientific claims, or interpret the experimental results. All AI-assisted text was reviewed and revised by the authors, and all AI-assisted code was manually inspected and validated through testing and experimental verification. The authors take full responsibility for the correctness, originality, claims, results, and final content of this work.