SIGHTLINES · VOL. 1, NO. 1 · NOVEMBER 2026 · THE SIGHT ACT · ORIGINAL RESEARCH

Quantization Matters

The error is in the edge.

The SIGHT Act Research Desk
One Frontier Foundation

NOVEMBER 2026 · RESEARCH CURRENT THROUGH NOVEMBER 17, 2026 · thesightact.org

ABSTRACT

The middle of a tensor is a solved problem. The edge is where the decisions are made. Quantization error is not distributed evenly across a block, a sequence or a benchmark. It concentrates where the distribution has its extremes, and those extremes are exactly the values the scale is built from. The mean sees the flat line. The buyer meets the edges.

An affine quantizer maps a real value onto a fixed grid. The scale, and the rule that picks it, is the format. A buyer who knows the precision can tell which variable moved. A buyer holding a price and a model name has no variables at all.

KEYWORDS Post-training quantization · endpoint disclosure · FP4 · block scaling · KV cache · calibration · black-box auditing

CONTENTS NINE SECTIONS · FOURTEEN SOURCES

1	The middle is cheap	2
2	The block maximum is the decision	3
3	0.1% of the values carry the distribution	3
4	The first tokens are an edge in time	4
5	Calibration memorises the edge	4
6	What the benchmark averages away	5
7	What a record would have to say	6
8	The audit at the edge	6
9	Two positions, not one	7
–	References	7

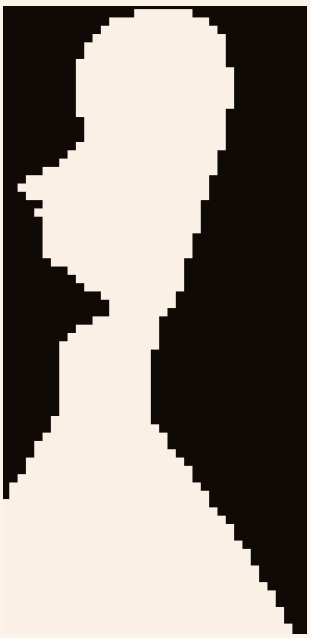


Plate 1. *Profile sampled at step 16.* The contour is the only visible error. Not to scale.

1 The middle is cheap

An affine quantizer maps a real value onto a fixed grid.

A quantized tensor is mostly correct. Rounding error, averaged over the interior, is small and roughly flat; a four-bit format reaches a perplexity close to full precision on a well-behaved model. That is the number the literature reports.

It is also the number that hides the failure. Quantization error is not distributed evenly across a block, a sequence or a benchmark. It concentrates where the distribution has its extremes, and those extremes are exactly the values the scale is built from. The middle of a tensor is a solved problem. The edge is where the decisions are made.

A quantizer maps a real value x to

$$\hat{x} = s \cdot Q(x/s) \tag{1}$$

where Q rounds to the nearest representable level (ties to even) and clamps to the element's range; the integer form

$$x_q = \mathrm{round}(x/s) + z \tag{2}$$

is the affine form (the z = 0 case is the symmetric quantizer), and FP4 has no zero point. Inside the range the error is bounded by half a step, so it falls as the step falls. In the E2M1 element that FP4 formats share, the codes are fixed: one sign bit, two exponent bits, one mantissa bit — sixteen codes, two zeros, and seven magnitudes in each sign (0.5, 1.0, 1.5, 2.0, 3.0, 4.0, 6.0). The grid is dense near zero and coarse at the top; the step between the two largest values is 2.0, and the step between 0.5 and 1.0 is 0.5.

That is the first edge: the element's own range. A scale that maps a block onto the top of the grid spends its resolution on values that occur once, and leaves the frequently-occurring middle coarse. A scale that maps the middle well clips the top. The choice of scale is the whole error budget, and it is set by one value per block: the maximum.

Table 1. The E2M1 magnitude grid in each sign: one sign bit, two exponent bits, one mantissa bit. Step is the difference to the next level; the two zeros share level 0. Source: NVIDIA, Introducing NVFP4.

LEVEL	VALUE	STEP
0	0.0	0.5
1	0.5	0.5
2	1.0	0.5
3	1.5	0.5
4	2.0	1.0
5	3.0	1.0
6	4.0	2.0
7	6.0	–

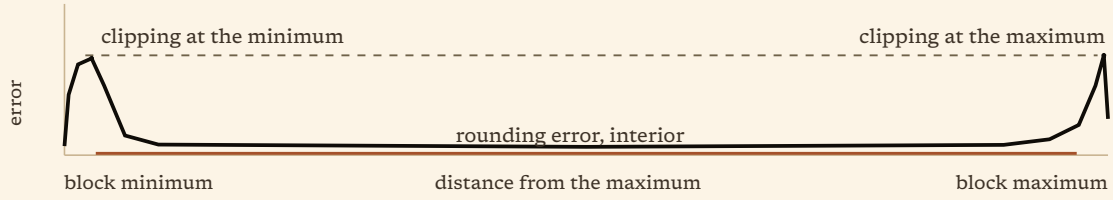


Figure 1. *The mean sees the flat line. The buyer meets the edges.* Rounding error is flat across the interior; the clay rule is that error, and the dashed line is the clip level. Clipping error is a function of the two ends of the block. Schematic, not to scale.

2 The block maximum is the decision

Every block-scaled format takes a maximum and derives a scale from it.

NVFP4 uses one FP8 E4M3 scale per sixteen values plus a per-tensor FP32 global; MXFP4 uses one E8M0 power-of-two scale per thirty-two values. The metadata cost differs — 4.50 bits per element against 4.25 — and so does the arithmetic: an E4M3 scale can sit between powers of two, while E8M0 is a power of two: under the OCP MX convention the scaled block maximum is placed in [4, 8) and clamps above 6.0.

NVIDIA's own comparison states the consequence for MXFP4's power-of-two grid: it "can create a large quantization error for the block maximum… and can often lead to larger overall quantization errors for blocks." In its worked NVFP4 example the average MSE is 0.08; the same text describes FP4 and MXFP4 as carrying a "risk of noticeable accuracy drop compared to FP8," and only NVFP4 as lower-risk.[1]

So the element type is not the whole format. The scale, and the rule that picks it, is the format. A seller who writes "4-bit" has named the container and left the decision out.

3 0.1% of the values carry the distribution

Model distributions are not Gaussian.

A small number of channels carry values far larger than the rest, and those values set the block maximum. Quantize the block with one scale and the outliers decide how much resolution every other value gets.

This is measured, not inferred. LLM.int8() found that emergent outlier features appear in large models at a rate of about 0.1% of all input features, and that these features are large enough to make per-tensor scaling fail; the method keeps them in FP16 and quantizes the remaining 99.9%. AWQ reaches the same conclusion from the other direction: protecting 0.1% of the weights — those salient by activation magnitude — recovers most of the lost accuracy. SmoothQuant migrates activation outliers into the weights before quantizing, because a per-tensor activation scale otherwise has to accommodate the maximum.

Three methods, one finding: the edge is a fraction of a per cent of the tensor, and it is most of the error.[2]

[1] NVIDIA, "Introducing NVFP4."

[2] T. Dettmers et al., "LLM.int8(): 8-Bit Matrix Multiplication for Transformers at Scale," arXiv:2208.07339, 2022, arxiv.org/abs/2208.07339 J. Lin et al., "AWQ: Activation-Aware Weight Quantization for LLM Compression and Acceleration," arXiv:2306.00978, 2023, arxiv.org/abs/2306.00978 G. Xiao et al., "SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models," arXiv:2211.10438, 2022, arxiv.org/abs/2211.10438.

4 The first tokens are an edge in time

A transformer's activation distribution is not stationary across a sequence.

The first tokens — reported attention sinks — take disproportionately large key and value magnitudes, and later tokens attend to them. Quantize the KV cache with one scale and the early positions dominate it.

The measured consequence appears in behaviour, not in perplexity. "Alignment Collapse Under KV Cache Quantization" covers eleven instruction-tuned models from 3.8B to 72B over five benchmarks and 1,894 prompts. Its headline: Mistral-7B loses 15.2% of its refusals at a perplexity ratio of only 1.03. Fifteen per cent of refusals, at a drift a perplexity gate would pass.

The same study finds no universal safe width. Qwen-2.5-7B shows a 90.3% conditional flip at 4-bit; Gemma-2-9B is clean through 3-bit and flips 91.8% at 2-bit; LLaMA-3.1-8B resists collapse even at a perplexity of 31.0. Within one nominal width, the FP8 format choice (2 vs 3 mantissa bits) is worth a 4× spread in damage: Qwen-2.5-7B under vLLM with FP8 KV cache flips 30.3% of baseline refusals under e5m2 and 7.1% under e4m3.

The mechanism is an edge effect. Key-only quantisation accounts for 76–102% of the alignment damage in eight of nine primary models, with key-projection MSE 4–87× higher than value-projection MSE. The values that matter are the first and largest, which is where the scale was set. The authors attribute part of the K-only excess to a WildGuard classification artifact.[3]

A separate vLLM report makes the same point without the alignment vocabulary: an FP8 KV cache at 128k context scored 13% against 91% for BF16, and a fix raised it to 89%. One kernel, one GPU generation, but the shape is the point — the error was at the long-context edge.[4]

Table 2. Alignment damage under KV-cache quantization. Refusals and conditional flips are measured against each model's full-precision baseline. Source: Alignment Collapse Under KV Cache Quantization.

MODEL	WIDTH / FORMAT	MEASURED EFFECT
Mistral-7B	4-bit KV	15.2% of refusals lost at a perplexity ratio of 1.03
Qwen-2.5-7B	4-bit	90.3% conditional flip
Qwen-2.5-7B	FP8 KV, e5m2	30.3% of baseline refusals flip
Qwen-2.5-7B	FP8 KV, e4m3	7.1% of baseline refusals flip
Gemma-2-9B	3-bit	clean through this width
Gemma-2-9B	2-bit	91.8% conditional flip
LLaMA-3.1-8B	3-bit	resists collapse at a perplexity of 31.0

5 Calibration memorises the edge

A quantizer needs data to choose scales.

GPTQ calibrates on 128 random segments of 2,048 tokens from C4 — 262,144 tokens — and the reference implementation defaults to `nsamples=128, seed=0` and hardcodes `seqlen=2048` in `llama.py`. Every reproduction on C4 fits the same 128 windows.

[3] B. C. Xu, A. Kumarappan, and M. Zhou, "Alignment Collapse Under KV Cache Quantization: Diagnosis and Mitigation," arXiv:2606.09864, 2026, arxiv.org/abs/2606.09864.

[4] J. Kübler, E. Kurtić, et al., "The State of FP8 KV-Cache and Attention Quantization in vLLM," vLLM Blog, April 2026, blog.vllm.ai.

When the evaluation text comes from the same distribution, the fit looks excellent. When it does not, the same weights tell a different story. GPTQ calibrated on C4 and scored on C4 versus PTB gives, for Sheared-LLaMA-1.3B, 9.34 against 1399.89 — a factor of about 150 on identical weights. LLaMA-13B gives 6.74 against 22.14. The ZeroQuant(4+2) authors title the section "GPTQ's Tendency to Overfit."

AWQ's defaults carry the same risk in a quieter form. The calibration helper's default is `n_samples=512, block_size=512` the CLI default is `n_samples=128, seqlen=512`. Same repository, same release: at most 65,536 tokens via the script, at most 262,144 via the API, and nothing at either call site says which produced a downloaded checkpoint. The calibration corpus is the Pile's validation split, so a downstream "Pile perplexity" is not held out from the fit.

Table 3. Calibration windows and the token counts they admit. The corpus travels with the checkpoint; the call site does not say which default produced it. Sources: GPTQ; AWQ.

METHOD / CALL SITE	CORPUS AND WINDOW	TOKENS (MAX)
GPTQ, reference	C4, 128 × 2,048	262,144
AWQ, helper	Pile validation, 512 × 512	≤ 262,144
AWQ, CLI	Pile validation, 128 × 512	≤ 65,536

The criterion matters as much as the corpus. NVIDIA's Model Optimizer group compared three ways of choosing per-block scales for the same NVFP4 format on Qwen3.5-9B: default max-scale lost 5.10 points against BF16 across four benchmarks, weight-tensor MSE lost 3.87, and output error under the local Hessian lost 3.10. A two-point spread from the scale rule alone — same bits, same block size, same element.[5]

6 What the benchmark averages away

A benchmark reports a mean over tasks.

The edge is a minority of tasks, so the mean moves little when the edge breaks.

The clearest example is instruction-following. A 13-benchmark study found quantized models beat smaller full-precision models on most tasks — except hallucination (TruthfulQA) and instruction-following (IFEval). Four-bit Llama-2-13B loses to FP16 Llama-2-7B on IFEval, and four-bit GPTQ Llama-3.1-405B at 202.5 GB underperforms FP16 Llama-3.1-70B at 140 GB on the same benchmark. The authors write that "quantization can compromise alignment and adherence to instructions."[6]

Safety behaves the same way. On Llama-2-7B-Chat, QLoRA at INT4 takes the baseline attack success rate from 0.3% to 85.3%, while MT-Bench sits at 6.40 and AlpacaEval at 63.92 — utility intact, guardrails gone. At 2-bit the same method reaches 82.0%.[7]

[5] E. Frantar et al., "GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers," arXiv:2210.17323, 2022, arxiv.org/abs/2210.17323 Lin et al., "AWQ"; X. Wu et al., "ZeroQuant(4+2): Redefining LLMs Quantization with a New FP6-Centric Strategy for Diverse Generative Tasks," arXiv:2312.08583, 2023, arxiv.org/abs/2312.08583 NVIDIA Model Optimizer Team, "Improving NVFP4 Accuracy with Local-Hessian Weight Scales," NVIDIA Model Optimizer, September 2026, nvidia.github.io.

[6] J. Lee et al., "Exploring the Trade-Offs: Quantization Methods, Task Difficulty, and Model Size in Large Language Models From Edge to Giant," arXiv:2409.11055, 2024, arxiv.org/abs/2409.11055.

[7] K. Chen et al., "Q-resafe: Assessing Safety Risks and Quantization-Aware Safety Patching for Quantized Large Language Models," arXiv:2506.20251, 2025, arxiv.org/abs/2506.20251.

7 What a record would have to say

The label "4-bit" survives every distinction in this issue.

A record that would let a buyer reconstruct the edge must carry, at minimum:

Table 4. The eight server-side facts a record must carry.

NO.	FIELD
1	the element format and its block size
2	the scale format and the scale-selection criterion (max, MSE, Hessian)
3	the clipping policy, and whether outliers are kept wide
4	which side is quantized: weights only, or weights and activations
5	the KV-cache format, its layer exemptions, and whether keys are quantized
6	the output-token cap and the reasoning cap
7	the calibration corpus and window length, where the checkpoint is third-party
8	the served revision, dated

All eight are server-side facts that a model name does not carry. The last is the one a provider can change without telling anyone; the others can drift between providers, as the postmortem in Section 9 shows.

8 The audit at the edge

The edge is also where black-box verification has power.

The Rank-based Uniformity Test needs one API call per prompt and a local reference model; against Gemma-2-9b-it and its own 4-bit variant it reaches power-AUC 0.392, where MMD reaches 0.214 and a KS baseline 0.017 – on budgets the source sets and does not equalise (Table 5). KBF, using knowledge-boundary probes and a 99% Clopper–Pearson bound, reports a 100% true-positive rate at a 1–6% margin across 155 model pairs. Its false positives appear only at the narrowest margins on 16 same-reference control retests: 2 of 16 at a 1% margin, 1 of 16 at 2–3%, none at 4–10%.[8]

Table 5. Black-box audit power at the edge: power-AUC, 500 trials at $\alpha = 0.05$. The budgets are the source's and are not equal: RUT and KS sample 100 prompts at 101 queries each, MMD 10 prompts at 10. The 8-bit pairs are the hard case. Source: Rank-Based Uniformity Test.

TEST	COMPARISON	POWER-AUC
RUT	Gemma-2-9b-it, 4-bit variant	0.392
MMD	Gemma-2-9b-it, 4-bit variant	0.214
KS	Gemma-2-9b-it, 4-bit variant	0.017
RUT	Gemma-2-9b-it, 8-bit	0.049
RUT	Mistral-7B, 8-bit	0.049
RUT	Llama-3.2-3B, 8-bit, WildChat	0.132
RUT	Llama-3.2-3B, 8-bit, MATH	0.253

[8] X. Zhu et al., "Auditing Black-Box LLM APIs with a Rank-Based Uniformity Test," arXiv:2506.06975, 2025, arxiv.org/abs/2506.06975 Y. Fang et al., "KBF: Knowledge Boundary as Fingerprint for Language Model and Black-Box API Auditing," arXiv:2605.29524, 2026, arxiv.org/abs/2605.29524.

The limit is the same edge, in the other direction: 8-bit substitutions are hard to see. RUT, MMD and KS all fail to separate the 8-bit variants of Gemma and Mistral in the published runs; RUT reaches only 0.132 on WildChat and 0.253 on MATH for the Llama-3.2-3B 8-bit pair, and KBF's quantized-backend verdict is a same-endpoint result. A label and an audit are complementary. The label covers what the audit cannot see; the audit covers what the label might be misstating. Neither substitutes for the other.

9 Two positions, not one

The series can hold two claims without resolving them.

i First: the label is necessary, because no black-box test in the published literature reliably separates an 8-bit endpoint from its reference.

ii Second: the label is insufficient, because a declared bit width does not predict behaviour. In a production postmortem covering roughly 18 million messages through OpenRouter, identical DeepSeek V4 Flash weights scored 90% GPQA first-party against 75% on DigitalOcean (81% to 58% on TAU), and a provider declaring fp4 scored 89.1% against a different provider declaring fp8 at 70.5%. The post's figures make the argument for disclosure rather than against it: a declared format did not predict the score.[9]

A buyer who knows the precision can tell which variable moved. A buyer holding a price and a model name has no variables at all.

References

FOURTEEN DATED SOURCES

K. Chen, J. Zhang, J. Hu, Y. Wang, J. Lou, Z. Feng, and M. Song. Q-resafe: Assessing safety risks and quantization-aware safety patching for quantized large language models. arXiv:2506.20251, 2025.
T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. arXiv:2208.07339, 2022.
Y. Fang, Y. Feng, B. Li, and M. Zhou. KBF: Knowledge boundary as fingerprint for language model and black-box API auditing. arXiv:2605.29524, 2026.
E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. arXiv:2210.17323, 2022.
J. Kübler, E. Kurtić, et al. The state of FP8 KV-cache and attention quantization in vLLM. vLLM Blog, April 2026.
J. Lee, S. Park, J. Kwon, J. Oh, and Y. Kwon. Exploring the trade-offs: Quantization methods, task difficulty, and model size in large language models from edge to giant. arXiv:2409.11055, 2024.
J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han. AWQ: Activation-aware weight quantization for LLM compression and acceleration. arXiv:2306.00978, 2023.
U. Malik. Debugging OpenRouter in production: The 10 provider bugs that bite. umesh-malik.com, September 2026.

[9] U. Malik, "Debugging OpenRouter in Production: The 10 Provider Bugs That Bite," umesh-malik.com, September 2026, umesh-malik.com.

NVIDIA. Introducing NVFP4 for efficient and accurate low-precision inference. NVIDIA Technical Blog, accessed November 2026. developer.nvidia.com.
NVIDIA Model Optimizer Team. Improving NVFP4 accuracy with local-Hessian weight scales. NVIDIA Model Optimizer, September 2026.
X. Wu, H. Xia, S. Youn, Z. Zheng, S. Chen, A. Bakhtiari, M. Wyatt, R. Y. Aminabadi, Y. He, O. Ruwase, L. Song, and Z. Yao. ZeroQuant(4+2): Redefining LLMs quantization with a new FP6-centric strategy for diverse generative tasks. arXiv:2312.08583, 2023.
G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. arXiv:2211.10438, 2022.
B. C. Xu, A. Kumarappan, and M. Zhou. Alignment collapse under KV cache quantization: Diagnosis and mitigation. arXiv:2606.09864, 2026.
X. Zhu, Y. Ye, T. Qiu, H. Zhu, S. Tan, A. Mannan, J. Michala, R. A. Popa, and W. Neiswanger. Auditing black-box LLM APIs with a rank-based uniformity test. arXiv:2506.06975, 2025.

DISCLAIMER

Views are the authors'; reasonable efforts were made, but accuracy is not guaranteed and errors and omissions are possible. Information only, not advice; product names identify only and imply no endorsement.

Original research from thesightact.org. Text, data and figures in this issue were assembled for the SIGHT Act's endpoint record and are published here first. Published November 17, 2026. Research current through November 17, 2026. SIGHTLINES · VOL. 1, NO. 1 · NOVEMBER 2026.