WHICH DECISIONS LOW-BIT QUANTIZATION BREAKS, AND HOW TO PREDICT THEM

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantization saves memory by storing model weights with fewer bits. It can also change model decisions, such as whether to call a tool or which option to choose from a finite set. We study these decision changes in 16 language models from 8 families at 4, 3 and 2 bits, across several post-training quantization settings. Our evaluation covers tool use, safety, general knowledge and social bias, using BFCL, XSTest, MMLU, BoolQ, BBQ and synthetic tasks. The decision *margin* is the score difference between two possible first tokens, measured before and after quantization. Its size shows how strongly the model prefers one option, and a change in sign means the preference has flipped. Writing the margin before quantization as m and the margin after quantization as m', we find an approximately linear relationship across decisions: $m' \approx c\,m + b$. The slope c is usually below one and becomes smaller as precision falls, so quantization progressively shrinks decision margins. The offset b is the same for every decision of one kind. Quantization therefore does not simply add random noise, and even a strong preference at full precision can flip. Quantization also affects different kinds of decisions to different degrees. Within tool use, whether to call a tool is often more sensitive than which tool to call: on 400 BFCL tasks, three of five models lose more completed calls than correct tool selections at 3-bit round-to-nearest. Under GPTQ and GGUF far fewer whether-to-call decisions flip than under plain rounding, so there is no single 3-bit failure point. The same relationship predicts how often decisions flip. Across 1,154 combinations of models, quantization settings, bit-widths and decision types drawn from our evaluation, we fit the slope, the offset and the spread around the fitted line on half of the decisions and predict the flip rate on the other half. The predicted flip rate differs from the observed flip rate by a median of 1.0 percentage point. Together, these results show a common pattern in how quantization changes decisions and provide a way to predict when those changes will occur.

1 INTRODUCTION

Post-training quantization reduces the memory required to run large language models (Dettmers & Zettlemoyer, 2023; Frantar et al., 2022), but it can also change the decisions they make. For tool-using agents, whether to call a tool and which tool to call are distinct decisions, and quantization need not affect them equally. After 3-bit rounding, Qwen3-4B can still choose the correct tool from a list of candidates. In the tool-calling test reported in §4.1, it produces no tool calls on requests that require one. We investigate how quantization changes such preferences and whether these changes can be predicted.

Quantization methods are designed around weight and layer-output error: Nagel et al. (2020) and Li et al. (2021) choose rounding values by their effect on a layer's output, and Lin et al. (2024) rescales salient weight channels. Neither error says which decisions will change.

We study 16 language models from 8 families at 4, 3 and 2 bits using round-to-nearest. We also evaluate three other quantization methods on all 16 models, and widths up to 8 bits on subsets of these models. Our evaluation covers tool use, safety refusal, general knowledge and social bias. Within tool use, we examine four choices separately: whether to call a tool, which tool to call, how to fill its arguments, and how to use its returned result.

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.

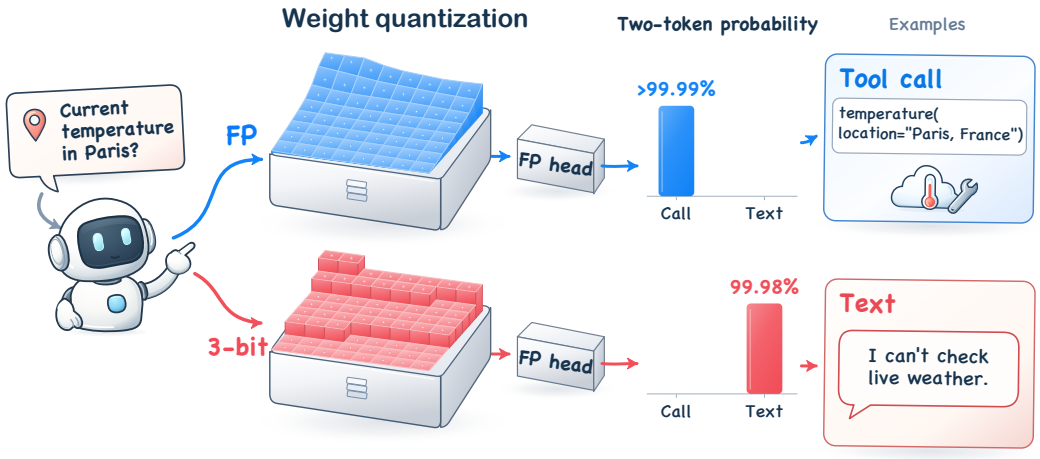

Figure 1: A tool-call preference reverses after 3-bit quantization. For the same weather request, Qwen3-4B favors the tool-call opening token at full precision (blue) and the text-opening token `I` after 3-bit rounding (red). The surfaces show sorted magnitudes of the same weight group before and after rounding. Both output heads remain at full precision. Bars show probabilities normalized over these two tokens only. Responses on the right are illustrative examples.

We study each *decision* as a choice between two options represented by distinct first tokens. Their *margin* is the difference between the model's scores (logits) for those tokens. A *flip* occurs when quantization reverses which token has the higher score. The *flip rate* is the share of readable items (two options starting with different tokens) that flip. Figure 1 illustrates one such reversal.

We find that quantization does more than add random noise to decision margins. Within one kind of decision, the quantized margin is approximately a linear function of the original: shrunk by a factor that falls as precision falls, and shifted toward one option. The shrinkage is also uneven across kinds of decision: at 3 bits under round-to-nearest, whether to call a tool collapses on most models, while which tool to call changes less.

In §5, we test whether this relationship predicts how often preferences reverse on new items. For each model, quantization configuration and kind of decision, we fit the relationship on half of the items. We then predict the fraction of the remaining items whose preferences will reverse. The forecast has a lower median error than using the fraction that reversed in the first half.

Contributions. (i) An evaluation across tool use, safety refusal, general knowledge and social bias. The evaluation shows that quantization disrupts whether to call a tool more than which tool to choose. (ii) An approximately linear relationship between margins before and after quantization: quantization often shrinks the original margin and shifts it toward one option. We test this relationship against three additive-noise models. (iii) A forecast of preference reversals, fitted on half of the items and evaluated on the other half. It beats the first half's observed flip rate.

2 Measurement and Experimental Design

2.1 Decision-level measurement

For the weather request in Figure 1, the model can open a tool call or start a text reply. We score the first token of each option on the same prompt, first at full precision and then after quantization.

Each *item* is a prompt x with two candidate tokens, t_1 and t_2. Their full-precision margin is

$$m(x) = \ell_{\mathrm{FP}}(t_1 \mid x) - \ell_{\mathrm{FP}}(t_2 \mid x), \tag{1}$$

where ℓ_{FP} is the token's score (logit). A positive margin favors t_1 and a negative one t_2. We compute the same difference after quantization, denoted by $m'(x)$. Each item thus gives one pair (m, m'). Opposite signs indicate a flip, and the flip rate is the fraction of readable items that flip.

For most tasks, t_1 begins the correct answer and t_2 begins the highest-scoring alternative at full precision. We keep this pair fixed after quantization. Whether-to-call instead compares calling with replying in text, and the two refusal tasks compare declining with complying. In these tasks, either option can be correct, and we record the correct option as the item's *side*. A flip can therefore add an error or fix one.

A margin measures preference between two tokens. Appendix A.1 checks agreement with generated tool calls and refusals. This correspondence weakens under severe quantization.

2.2 Experimental design

Benchmarks and decisions. Four test sets hold 1,092 items from benchmark subsets and authored tasks, covering fourteen kinds of decision. Tool calling is the main case: BFCL (Yan et al., 2024) supplies whether-to-call requests, while authored tasks isolate tool selection, argument filling and tool-result use. A separate 400-task BFCL evaluation measures call completion and tool selection on the same requests. Its five models pass the full-precision capability checks specified in Appendix A.19. Public which-tool (BFCL, MetaTool Huang et al., 2024) and argument-fill (BFCL) tasks repeat two decisions in Table 1. To examine decisions beyond tool use, we use XSTest for refusal (Röttger et al., 2024), MMLU and BoolQ for knowledge and question answering (Hendrycks et al., 2021; Clark et al., 2019), and BBQ for social bias (Parrish et al., 2022). Synthetic arithmetic and code tasks add procedural choices, including *code tokens*, the choice of the next symbol in a short piece of code. Appendix A.6 gives sampling, counts and token pairs.

Models. The weight experiments cover 16 models from eight families: Qwen2.5, Qwen3, Qwen3.5, Granite, Gemma-3, Gemma-4, Llama-derived models and Phi. Model sizes range from 0.6B to 32B. Four additional base-model controls test whether the pattern also occurs before instruction tuning.

Quantization and controls. Weight quantization targets decoder-block linear layers. Embeddings and output heads remain at full precision. We evaluate:

- **Round-to-nearest (RTN):** 4, 3 and 2 bits on all 16 models, with 8-, 6- and 5-bit sweeps on five models.
- **Activation-aware scaling:** 4, 3 and 2 bits on all 16 models.
- **GPTQ:** 4, 3 and 2 bits on all 16 models.
- **GGUF:** nominal 8-, 4-, 3- and 2-bit builds of all 16 models.

Additional experiments vary activation and KV-cache precision, test rotation and repairs, and check alternative token pairs and margin-matched items. Vision–language and governance tasks test the measurement's validity in additional domains. Appendix A.6 gives the full quantization coverage.

Analysis protocol. A *condition* is one model configuration, bit-width and kind of decision in one run. Calling and refusal are also analyzed by side. Appendix A.21 details the counting. We fit on half of the items and predict flips on the other half, also checking a split that keeps repeated requests together. Section 5 reports the forecast, baseline and cross-model transfer comparisons.

3 The damage is a fraction, not an amount

What the plot shows. Figure 2 puts quantized margins against full-precision ones, and it passes through three stages. At 4 bits the points sit on a line whose slope is a little below one: everything has shrunk by about a sixth and shifted down by three logits, and almost nothing has changed sides. At 3 bits the line is nearly flat and most of what was positive is now negative. The two sides separate into clusters there. The separate fits in Appendix A.18 give low slopes on both sides, so the pooled

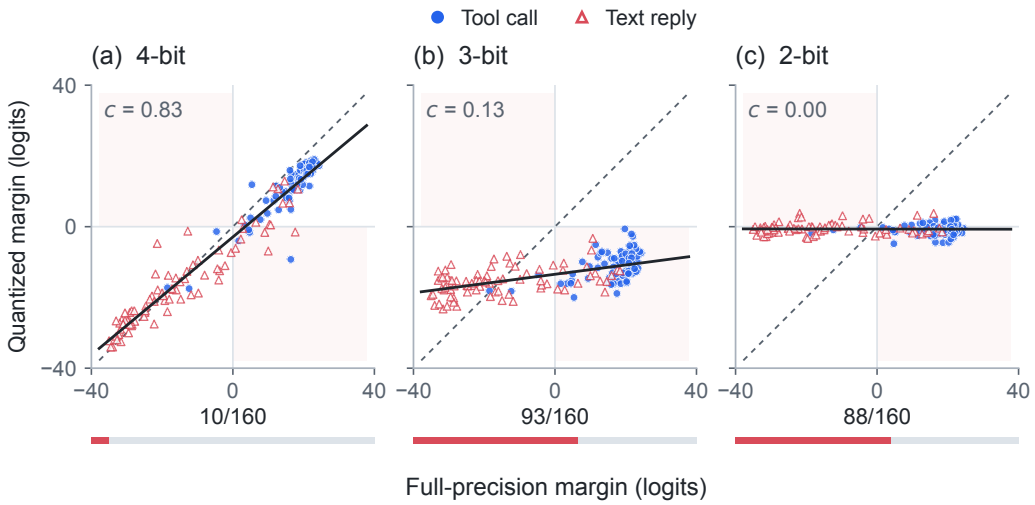

Figure 2: Quantized against full-precision margins (Qwen3-4B, whether-to-call, round-to-nearest). All three panels use identical square axes. Dashed lines are the identity, solid lines are fitted responses, and pale quadrants indicate sign changes. Colour gives each item's correct response, its should-call or should-not-call side. The counts and strips below report flips among the same 80 items a side. Labels give fitted slopes c. Against the additive account, the cloud tips toward the horizontal instead of keeping the identity's angle.

slope is shrinkage within each side and not the gap between them. At 2 bits the line is flat: the fitted slope is zero.

Additive noise would give a band of constant width along the diagonal. The data give a line with a slope below one and an intercept away from zero. We write the fitted relation as

$$m' = c \cdot m + b + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2). \tag{2}$$

c is the *surviving fraction*: 1 when nothing is lost, 0 when the quantized margin no longer depends on the original. Below zero, the quantized margin points the opposite way to the original. One condition in six where c can be estimated has it below zero. Over all the measurements in this paper its median is 0.86 at 4 bits, 0.33 at 3 and 0.00 at 2. The *offset* b moves every margin in the same direction. On tool calls the offset is a penalty on the one token that opens the call, which §4.1 shows by scoring the whole vocabulary on Qwen3-4B. A drift shared by every logit, such as the bias quantization is known to add to the weights (Nagel et al., 2021), would cancel in a margin. σ is the residual spread. All three are fitted per condition. We call Eq. 2 *the law*, and test it below against three additive alternatives. Flips come from all three terms: shrinkage removes the protection a large margin gave, the offset decides which way the decision falls, and the residual flips the smallest margins either way.

Appendix Figure 7 shows that the surviving fraction falls steeply between 4 bits and 2 in all eleven models where it can be estimated. The model is not simply turning its volume down: rescaling every logit by one factor would change no decision. The damage is also uneven: on Qwen3-4B under 3-bit round-to-nearest, the surviving fraction is 0.87 for code tokens against 0.13 for whether-to-call (every kind is in Table 1).

Ruling out the additive story. An additive model whose noise grows with the margin draws the same picture as a slope below one. So we fit four competing accounts by maximum likelihood and let BIC pick between them. BIC rewards a better fit and charges for every extra parameter. Two of them are constant-size noise, with and without a drift term. The third is additive noise whose size grows with the margin, which we call *margin-scaled noise*: the only additive form that widens its spread with the margin without being a special case of the law. The fourth is the law itself.

A condition is *damaged* when its fitted c is below 0.70. That definition is about how much of the margin is left, not about whether any decision changed sides: a kind of decision whose margins are wide can lose most of them and still flip nothing. The multiplicative account wins 651 of the 746

Table 1: Every kind of decision at 3 bits, under four quantizers. Each cell gives two medians: the surviving fraction c over the models where it can be estimated, and the share of decisions that change sign (%) over all models tested. A superscript gives how many models were tested on that decision when fewer than 16 were; – means that fewer than four models give a c with standard error at most 0.10. GGUF margins are read against the same model's f16 conversion. Activation-aware scaling is scored on the items not used to calibrate it. The arithmetic row uses 300 items. BFCL and MetaTool rows are public tasks (Appendix A.6). Appendix Table 6 gives 4, 3 and 2 bits.

	Round-to-nearest		Activation-aware		GPTQ		GGUF iq3_xxs	
Decision	c	flip	c	flip	c	flip	c	flip
Tool use								
Whether to call	0.22	41	0.34	39	0.55	21	0.71	10
Which tool, ours	–	0	–	0	–	0	–	0
Which tool, BFCL	0.25	11	0.46	5^{15}	0.59	2^{12}	0.70	0^{14}
Which tool, MetaTool	0.24	16	0.48	11^{15}	0.63	6^{12}	0.76	5^{14}
Argument fill, ours	0.61	0	0.89	0	0.85	0	0.90	0
Argument fill, BFCL	0.59	4^{15}	0.67	2^{15}	0.71	4^{6}		
Tool-result use	0.38	0^{15}	0.42	0^{7}	0.77	0^{11}	0.91	0^{11}
Refusal								
Forced choice	0.25	35	0.35	23	0.55	14	0.76	6
Open-ended opener	0.40	7	0.54	3	0.70	4	0.77	2
Knowledge								
MMLU	0.19	35	0.37	32	0.54	27	0.73	18
BoolQ	0.49	23	0.50	21	0.63	12	0.87	6
Procedure								
Arithmetic	0.37	22	0.39	6	0.63	5	0.59	1
Code token	0.83	10	0.88	4	0.91	10	0.99	10
Social bias								
BBQ	0.26	40	0.41	32	0.52	22	0.76	15

damaged conditions. This comparison extends beyond the four main test sets to include governance and vision–language tasks, the external tasks of §4.1 and its margin-matched controls. Appendix Table 9 lists the data included in each analysis. The cutoff sensitivity checks in Appendix A.5 leave the verdict unchanged. The multiplicative account wins every whether-to-call condition and every refusal condition, forced-choice and open-ended. Neither constant-noise account wins a damaged condition. Margin-scaled noise takes the remaining damaged conditions. Appendix A.22 repeats the comparison by severity, and the verdict holds in every band of damage.

One kind of decision is an exception. Tool-result use, reading a returned tool value back out of the context, prefers the margin-scaled-noise account in all but one of its damaged conditions. Testing more candidate laws shows what fails: the shrinkage is still multiplicative, and only the constant-spread assumption fails (Appendix A.5).

4 Which decisions break first, and where the break moves

The surviving fraction varies with bit-width, kind of decision, model and what else is quantized, so it has to be measured on the model being shipped. This section says which decisions go first (§4.1, Table 1), where the breaking point moves (§4.2) and what repairs the damage (§4.3). Benchmark accuracy alone does not show these thresholds (Appendix A.9).

4.1 Whether to call a tool collapses before which tool to call on most models, and we say on which it does not

At the bit-width where damage begins, the two halves of a tool-calling decision separate. *Which* tool to call moves less, and on the items we wrote hardly at all. Take the 34 model-and-quantizer runs in Appendix Figure 8, each measured at 3 bits on both decisions. In 32 of them not one which-tool decision changes side, against a median whether-to-call flip rate of 0.48. The which-tool test has

eight distinct requests, each scored five times against the same pair of options, so a zero rests on eight measurements and not forty. Appendix A.6 specifies how requests and options are paired. *Whether* to call collapses, mostly toward not calling. Where quantization damages whether-to-call without destroying it, the model errs toward not calling about two thirds of the time: 68 of the 101 such conditions lean toward not calling. Margins on the should-call side change sign at nearly twice the rate of the other side (Appendix Table 9 gives the filter and each tally).

The condition the rest of the paper follows, 3-bit round-to-nearest on Qwen3-4B, is one of the conditions that saturate, flipping 96% of should-call against 20% of should-not-call (Appendix Table 7; the other five models measured the same way are in Appendix Table 8). A model that stops calling flips every item it used to call: most should-call items, and the few should-not-call items it called wrongly at full precision. The offset predicts how far a condition leans. Conditions whose offset is small next to their spread sit near even, and conditions whose offset is several spreads wide lean hard. Each model leans in the direction of its own fitted offset, and a minority lean toward calling (Appendix A.18).

Scoring the whole vocabulary says what the offset is made of. At 3 bits the token that opens a tool call sits at the very bottom of the vocabulary-wide change in logits on three models. The call token also sits far below the other reserved tokens, and below ordinary tokens matched to the call token on the size of their output-layer row. These vocabulary-wide and matched-token comparisons are reported in Appendix A.15. Both comparisons support a penalty on the call token rather than a vocabulary-wide drift or a token-class effect. Appendix Figure 8 shows whether-to-call and which-tool parting at 3 bits and both failing at 2.

The tool test set carries four kinds of decision, and the contrast so far uses two. On the 24 builds where all four were measured in the same run, whether-to-call changes side at 3 bits on every one of them. Filling an argument and reading a tool result across turns sit at a median of zero, as which-tool does. Argument filling on public BFCL tasks flips a median of 4% at 3 bits under round-to-nearest. The tool-result rate is read only over the items whose two answer options begin with different tokens. After rewriting the rest so they do and re-measuring, tool-result use still does not flip at 3 bits (Appendix A.18). Counting each model and quantizer once rather than once per run (the per-model row of Appendix Table 9), whether-to-call flips more than which-tool in 19 of the 20 model-and-quantizer pairs measured on both at 3 bits. The one exception is a no-zero-point variant of round-to-nearest on Qwen3-4B, where which-tool flips everywhere. Two conditions in the main evaluation move which-tool at 3 bits and both are on that model: this one, and one of five stochastic-rounding replicates of plain round-to-nearest. On public tasks which-tool does change, but less: at 3 bits under round-to-nearest it flips a median of 11% on BFCL and 16% on MetaTool, against 41% for whether-to-call, and less than whether-to-call on 13 of 16 models on BFCL and 13 of 16 on MetaTool. Appendix A.18 gives the four-kind comparison and both exceptions.

Four checks test whether the whether-to-call/which-tool gap is an item or token effect, and Appendix A.18 gives each design in full. Within a condition the which-tool items all start with nearly the same margin, and at 2 bits they lose that margin like every other kind. Pairing items across the two kinds on their full-precision margin gives two dozen matched pairs on Qwen3-4B, and at 3 bits whether-to-call changes side on most of them and which-tool on none. Rescoring the same prompts against other text openers leaves the bit-width of the collapse where it was on every model tried, and changes only how large the collapse is. Ten which-tool items whose two tools are near-synonyms have far less margin to spare than ten whether-to-call items whose request needs the tool, and at 3 bits the large-margin whether-to-call items flip while most of the small-margin which-tool items hold, so margin size is not the cause.

Those results come from items we wrote. We therefore put both decisions on the same 400 tasks from the public tool-calling benchmark BFCL (Yan et al., 2024), with the test fixed before the runs. There whether-to-call is completing the call and which-tool is naming the right tool, and for each we measured the share of tasks the full-precision model gets right and the quantized one does not (Appendix A.19). Five models answer well enough at full precision to be worth quantizing. Of the 400 tasks, 80 are used to fit c, b and σ and 320 are held out. Every loss below is read on the held-out tasks the full-precision model got right on both the call and the name, 260 to 300 per model. On three of the five, Gemma-3-4B, Qwen2.5-7B and Qwen3-4B, the completed call goes first by a clear lead at 3-bit round-to-nearest (Appendix Table 11). The three do not lose the call the same way: Gemma-3-4B and Qwen3-4B stop opening it, while Qwen2.5-7B still opens it every time and fails to

finish writing it, so on that model the decision held and the format broke. Averaging the five builds equally gives a gap of 20.2 percentage points between the two losses, and resampling whole tasks puts its lower bound at 16.9. Before running any of this we fixed what would count as a result. At least three builds had to show a gap of 15 percentage points or more, and the average over the builds had to have a lower bound above 10. Both hold. Two models do not lose the call first, and they set the scope of the claim. Hermes-3-Llama-3.2-3B loses the name first, and its lost names go to another listed tool rather than to broken output, so its preference moved rather than its formatting. Qwen3-1.7B loses both at once: nearly all of its output names no tool, the total collapse the other models show at 2 bits. The margins say why the call goes first. On Gemma-3-4B at 3 bits both decisions lose nearly all of their margin while their offsets separate, a penalty on the call token and none on the name, which is the mechanism of §3 on external items.

4.2 Refusal margins go one bit lower, and activations are a second axis

Under round-to-nearest, refusal margins are still mostly intact at 3 bits on models of 4B and above, where whether-to-call margins have collapsed, and they go at 2. We ask for refusals two ways, a forced choice and an open-ended opener. At 4 bits the current-generation instruct models of 4B and above keep their refusal behaviour under both. At 3 bits, on the forced choice (the more sensitive of the two), model size mostly decides. Models below 4B lose most of the refusals they made at full precision, models of 4B and above lose few, and one 9B breaks the size order. At 2 bits the size order is gone and roughly half the refusals go in every model but one. Appendix A.8 gives the model-by-model refusal results, including these exceptions and the generated-response check below. Decoded behaviour follows the margins at 4 bits and runs ahead of them at lower widths. In free completions from Qwen3-4B the refusal rate is unchanged at 4 bits, falls by a third at 3 bits while the first-token margins still prefer refusal, and at 2 bits the text neither refuses nor complies. Chhabra & Khalili (2025) report that the internal direction carrying refusal survives quantization at 4 bits.

The advertised bit-width does not fully describe the quantization configuration. Quantizing the activations as well as the weights is a second axis with its own breaking points. Activations are the harder axis (Lin et al., 2025), and we add where those points land per kind of decision. The standard fix is a block-diagonal Hadamard rotation applied before rounding, which spreads the outliers that make activations hard to quantize. We call it *the rotation*. Appendix Figure 5 plots weight width against activation width, five models from four families. Eight-bit activations are effectively free and combine with four-bit weights. Four-bit activations at the linear-layer inputs, under the plain per-token quantizer, destroy the decision margin **whatever the weights are doing**. The rotation moves that collapse about three bits lower rather than removing it. At 4-bit activations, how much margin survives depends on the model: of the four models measured plain and rotated, on two the margin comes back most of the way and on the other two hardly at all. Appendix A.13 compares the plain and rotated schemes on each model. Which activations are quantized matters more than how many bits they get. At 4-bit activations with 8-bit weights, quantizing only the KV cache flips 3 to 18% of decisions on the four of those models where it was measured, on whether-to-call and the forced-choice refusal. At the same widths, quantizing the inputs to the linear layers flips 44 to 84% and collapses the surviving fraction.

4.3 What repairs the damage, and what only looks like it does

Two choices about where bits go move the surviving fraction: the width of the **KV cache**, the memory an agent carries across a long context, and the rotation of §4.2. At equal nominal width the cache is a more sensitive place to lose bits than the weights, though less so than the linear-layer inputs. On the two models we swept fully, 4-bit weights with a full cache leave most of the margin, while full weights with a 4-bit cache remove a fifth to two fifths of it. The cost falls on the decisions weight quantization hits hardest (Appendix A.14). No cache width below 8 bits is safe for every model (compare Xu et al. 2026). How much a cache width costs depends on the quantizer, not on the width alone. Two cache schemes differing only in rounding convention differ by up to half of the surviving fraction at the same width. A cache result that does not name its scheme is not reproducible.

The label-free repairs we tested are per-item offsets, three definitions of "important" weights, leaving the last blocks at full precision, rounding each weight by the margin's gradient, adding noise before rounding (dithering), and ordering layers by how much the margin depends on each. Each is set

against its control in Appendix A.10. None wins on a majority of the models it was tried on, and none comes near the 30-point fall in flip rate that one more bit brings to a severely damaged condition. Of the three importance definitions, margin sensitivity nearly doubles the surviving fraction while its flip rate stays put (Appendix A.10). The equal-byte comparison in Appendix A.14 finds that attention-only protection wins on three of seven models and loses on four. One loss is smaller than the estimation noise floor defined in Appendix A.4. One class is excluded by the algebra of §3. Any repair of the form $\ell \to \alpha\ell + \beta$ with a single $\alpha > 0$ and a single β leaves every sign where it was. This algebra rules out temperature scaling, a global bias offset, and the re-fitted logit scale of Bouguerra et al. (2026), whatever that does for calibration.

5 What you have to measure yourself

Everything so far was measured on builds we quantized ourselves. A deployer fits c, b and σ once on the model being shipped instead of reusing ours.

The relation predicts data it was not fitted on. Fix a kind of decision and a way of quantizing, and suppose the quantized margin is Gaussian around $c\,m + b$ with a spread that does not depend on m. The chance that a decision flips is then a normal tail, and with the measured c and b it is

$$P(\text{flip} \mid m) = \Phi\big(-(cm + b)\,\text{sign}(m)/\sigma\big) \tag{3}$$

with Φ the standard normal CDF. Fitted on half of a condition's items and scored on the other half, the forecast predicts the flip rate to a median of 1.0 percentage point over the 1,154 conditions with enough readable items to split this way, 717 of them from additional experiments. The forecast is least accurate where there is most room to be wrong, on the conditions whose observed rate sits mid-range, as Figure 3 shows. The forecast also beats simply reusing the fitting half's own flip rate. On the conditions whose observed rate lies between 0.2 and 0.8 the law's median error is 4.3 percentage points against 6.0, and per decision its predicted probabilities sit a quarter closer to what happened, in squared error. Over all 1,154 conditions the law is the closer of the two on 46%, the fitting half's own rate on 33%, and the rest are level, which is where nothing flips on either half. Appendix A.20 gives the full baseline comparison and checks a split that keeps repeated requests together. Constants copied from another model miss by up to 33 percentage points at 3 bits on the two models tried, four times the worst 4-bit miss. The forecast predicts a change of the margin's sign, not of the model's output. Predicting which calls a model will fail to emit is validated only on Qwen3-4B at 4 bits. On every other build, including every 3-bit build, it does not predict which calls go missing in the BFCL generation test of Appendix A.19.

6 Conclusion

Which decisions break depends on their kind: at 3 bits under round-to-nearest, whether to call a tool collapses on most models while which tool to call changes less, and refusal margins hold until 2 bits on models of 4B and above. A deployer can do one paired pass on the model and width being shipped: run the decisions once at full precision and once quantized, and fit the surviving fraction, the offset and the residual spread to the two readings. The pass forecasts the rate of sign changes wherever the surviving fraction can be estimated. Where it cannot, report the observed sign-change rate.

Four-bit activations into the linear layers destroy the margin whatever the weights keep, and the rotation only moves that collapse lower. A 4-bit cache costs more of the margin than 4-bit weights do. Outside text the question is open: Appendix A.16 shows why our vision–language test cannot settle it.

7 Limitations

What this covers. Post-training quantization only. Decisions are read at the first token, and how damage accumulates over a rollout is not covered. Sixteen models have their weights quantized. Five have their activations quantized: four of the sixteen and Qwen3-8B, which is on the activation axis only. One vision–language family covers the second modality. Every weight run quantizes the linear

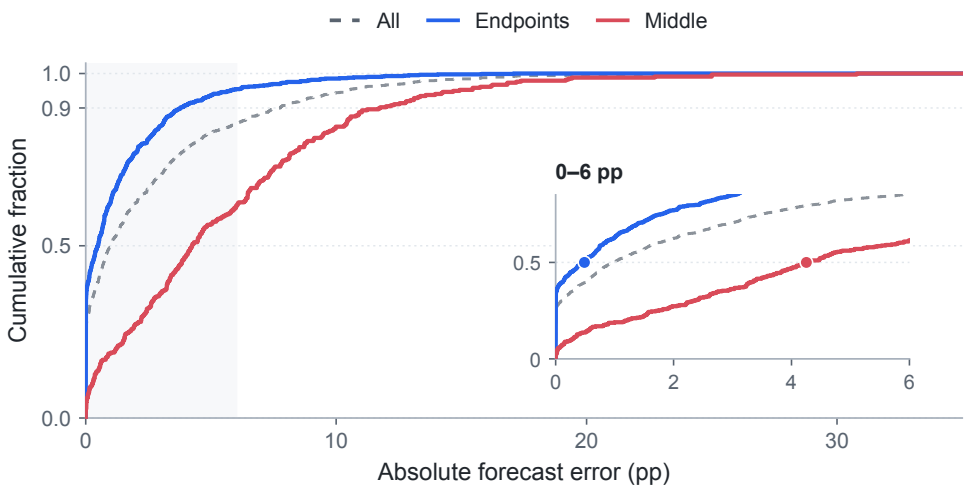

Figure 3: Held-out forecast errors across the 1,154 conditions. The full CDF includes a magnified view of errors up to 6 percentage points. Circles in the inset mark the endpoint and middle medians. Overall the median and 90th percentile are 1.0 and 7.7 points; on the 337 conditions whose observed flip rate r lies between 0.2 and 0.8, they are 4.3 and 11.7. The other 817 conditions lie at or near an endpoint ($r \leq 0.2$ or $r \geq 0.8$). Their errors are smaller throughout.

modules inside the decoder blocks and leaves the embedding table at full precision, so a build we call 4-bit is above 4 bits in effective terms. Appendix A.6 gives the coverage census.

Two limits of scope. The 4-bit refusal statement holds for standard current-generation instruction-tuned builds of 4B and above. Outside that set a model can lose a large share of the refusals it makes at full precision, at 4 bits up to two in five and at 3 bits nearly all (Appendix A.8). Gemma-4-E4B is left out of the counts on the activation axis because it keeps 35.5% of its parameters at full precision, which makes its series a weaker intervention than the other four (Appendix A.6). Its activation conditions still enter the held-out forecast, and dropping them leaves the median error where it is.

Mechanism. We describe the damage (a call-token penalty) but do not explain it, and why large models keep more margin is open.

What the external test does not settle. The break order and the missed-call forecast were tested on one external task set and five models. Four narrower limits are in Appendix A.7.

Ethics Statement

One measurement in this paper is dual-use: 3-bit quantization removes most safety refusals on models below 4B and a third of Qwen3-4B's in free generation, and an aggregate accuracy score does not separate that loss from ordinary compression damage (§4.2; Appendix A.9). Deliberately harmful quantization is already published (Egashira et al., 2024), and the removal we measure is what standard post-training quantization does on its own. The hazard is therefore present whenever a deployer compresses a model and validates it on overall accuracy. A precise map of which refusals go first may still lower the cost of finding them, and we accept that. The audit surfaces the loss of the refusal preference at the first token. The decoded refusal a user reads falls at 3 bits while that preference still holds on some models, and the responses that lose it mostly deflect the request rather than comply (Appendix A.8), so first-token preference, decoded refusal and compliance are reported separately. All models and quantizers are public. The evaluation items, margins and scripts will be released with the code. The test sets contain no personal data, and no human subjects were involved.

AI USE STATEMENT

Large language models were used throughout this work, and the disclosure is by category so that it can be checked. Code: the measurement, analysis and figure scripts were written with model assistance and reviewed by the authors. Experiments: models helped launch, monitor and collect the runs. Every measurement is a model forward pass recorded in a named result file. Data: the external tasks are drawn from a public benchmark unchanged. The safety requests are XSTest's (Röttger et al., 2024), wrapped in prompts the authors wrote; the tool items that are not BFCL's were written by the authors. Ideas: the form of Eq. 2 and the hypotheses tested are the authors'; models were used to argue against them and to search the literature. Analysis: models ran much of the analysis and proposed readings of the results, including several that were adopted and several that were wrong. The authors checked each against the stored files and decided what the paper claims. That loop caught two wrong claims and corrected which conditions the headline forecast covers. Writing: models drafted and revised prose, which the authors edited. Verification: two automated checks run over the source. One lists 516 claims, each tied to a result file and a phrase in the text, and fails if the stored value is absent there. A claim not on the list is not checked by it. The other reads the draft against itself: a quantity written two ways in two places, a figure whose caption and text disagree or that is printed at a scale its labels cannot survive, a term used before the paper marks its definition. Neither check reads the paper's reasoning, and model-assisted analysis produced at least one error that reached a draft before a check caught it. The authors have read every claim that remains and are responsible for all of them.

REPRODUCIBILITY STATEMENT

Every number in this paper is produced by a checked-in script from a named result file. The listed-claim check is the one described above. Numbers outside its list are checked by hand. Appendix A.6 gives the item counts, sources and build tags, Appendix A.4 the estimator and its floor, and Appendix A.20 the split protocol. Code, test sets, and the full result matrix will be released.

REFERENCES

Mohammad Asadi, Jack W. O'Sullivan, Fang Cao, et al. Mirage: The illusion of visual understanding. *arXiv preprint arXiv:2603.21687*, 2026.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. *Advances in Neural Information Processing Systems*, 2024.

Abhimanyu Bellam and Jung-Eun Kim. Explaining how quantization disparately skews a model. *arXiv preprint arXiv:2509.07222*, 2025.

Aymen Bouguerra, Daniel Montoya, Alexandra Gomez-Villa, Chokri Mraidha, and Fabio Arnez. Less precise can be more reliable: A systematic evaluation of quantization's impact on VLMs beyond accuracy. In *International Conference on Machine Learning (ICML)*, 2026.

Ting-Yun Chang, Muru Zhang, Jesse Thomason, and Robin Jia. Why do some inputs break low-bit LLM quantization? *arXiv preprint arXiv:2506.12044*, 2025.

Vishnu Kabir Chhabra and Mohammad Mahdi Khalili. Towards understanding and improving refusal in compressed models via mechanistic interpretability. *arXiv preprint arXiv:2504.04215*, 2025.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2019. URL `https://arxiv.org/abs/1905.10044`.

Tim Dettmers and Luke Zettlemoyer. The case for 4-bit precision: k-bit inference scaling laws. In *International Conference on Machine Learning (ICML)*, 2023.

Hui Dong, Yanzhao Li, Jie Gao, Chunlu Li, Zhiyuan Zhang, Yupeng Sun, Zhenyuan Chen, and Zhiqiang Zou. HiFA4: Training-free 4-bit FlashAttention on Ascend HIF4 NPUs for LLM inference. *arXiv preprint arXiv:2607.04302*, 2026.

Peijie Dong, Zhenheng Tang, Xiang Liu, Lujun Li, Xiaowen Chu, and Bo Li. Can compressed LLMs truly act? an empirical evaluation of agentic capabilities in LLM compression. *arXiv preprint arXiv:2505.19433*, 2025.

Abhinav Dutta, Sanjeev Krishnan, Nipun Kwatra, and Ramachandran Ramjee. Accuracy is not all you need. *arXiv preprint arXiv:2407.09141*, 2024.

Kazuki Egashira, Mark Vero, Robin Staab, Jingxuan He, and Martin Vechev. Exploiting LLM quantization. *Advances in Neural Information Processing Systems*, 2024. arXiv:2405.18137.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2009.03300.

Yue Huang, Jiawen Shi, Yuan Li, Chenrui Fan, Siyuan Wu, Qihui Zhang, Yixin Liu, Pan Zhou, Yao Wan, Neil Zhenqiang Gong, and Lichao Sun. MetaTool benchmark for large language models: Deciding whether to use tools and which to use. In *International Conference on Learning Representations (ICLR)*, 2024.

Jiwon Jang, Kisu Yang, Heuiseok Lim, and Hyunwoo Park. Flat score, amplified failures: How the error budget masks damage in quantized LLM agents. *arXiv preprint arXiv:2607.27275*, 2026.

Mingyu Jin, Kai Mei, Wujiang Xu, Mingjie Sun, Ruixiang Tang, Mengnan Du, Zirui Liu, and Yongfeng Zhang. Massive values in self-attention modules are the key to contextual knowledge understanding. In *International Conference on Machine Learning (ICML)*, 2025.

O. M. Kiselev. Boundary-aware quantization: Finite-scale decision geometry of neural classifiers. *arXiv preprint arXiv:2607.01478*, 2026.

Tanishq Kumar, Zachary Ankner, Benjamin F. Spector, Blake Bordelon, Niklas Muennighoff, Mansheej Paul, Cengiz Pehlevan, Christopher Ré, and Aditi Raghunathan. Scaling laws for precision. In *International Conference on Learning Representations*, 2025. arXiv:2411.04330.

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. In *International Conference on Learning Representations (ICLR)*, 2021.

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. In *Proceedings of Machine Learning and Systems (MLSys)*, 2024.

Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. QServe: W4A8KV4 quantization and system co-design for efficient LLM serving. In *Proceedings of Machine Learning and Systems (MLSys)*, 2025.

Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? Adaptive rounding for post-training quantization. In *International Conference on Machine Learning (ICML)*, 2020.

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021.

Xu Ouyang, Tao Ge, Thomas Hartvigsen, Zhisong Zhang, Haitao Mi, and Dong Yu. Low-bit quantization favors undertrained LLMs: Scaling laws for quantized LLMs with 100t training tokens. In *Proceedings of ACL*, 2025. arXiv:2411.17691.

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel R. Bowman. BBQ: A hand-built bias benchmark for question answering. In *Findings of ACL*, 2022.

Irina Proskurina, Luc Brun, Guillaume Metzler, and Julien Velcin. When quantization affects confidence of large language models? In *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024.

Baha Rababah, Cuneyt Gurcan Akcora, and Carson K. Leung. The illusion of equivalency: Statistical characterization of quantization effects in LLMs. *arXiv preprint arXiv:2607.08734*, 2026.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2024.

Samuel Salfati. Quantization dominates rank reduction for KV-cache compression. *arXiv preprint arXiv:2604.11501*, 2026.

Bruce Changlong Xu, Adarsh Kumarappan, and Mu Zhou. Alignment collapse under KV cache quantization: Diagnosis and mitigation. *arXiv preprint arXiv:2606.09864*, 2026.

Fanjia Yan, Huanzhi Mao, Charlie Cheng-Jie Ji, Tianjun Zhang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Berkeley function-calling leaderboard. `https://gorilla.cs.berkeley.edu/leaderboard.html`, 2024.

Chenxi Zhou, Pengfei Cao, Jiang Li, Bohan Yu, Jinyu Ye, Jun Zhao, and Kang Liu. Task-stratified knowledge scaling laws for post-training quantized large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026.

A Appendix

The main text, continued

The passages below continue the main text in full, in reading order.

A.1 What the instrument reads, and where it stops

The margin instrument scores two designated tokens. Here we compare their preference with generated responses and identify where the correspondence fails.

The worked example in its population. The whether-to-call set on Qwen3-4B has 80 should-call items and 80 should-not-call ones. At full precision the margins have the intended sign on 77 of the 80 should-call items and on 64 of the 80 should-not-call ones. The following transitions use those full-precision-correct items. The should-call median is $+18.75$ at full precision, $+13.84$ at 4 bits and -11.56 at 3, while the should-not-call median stays negative throughout. Counting how many of the 77 correct calls changed side: 4 at 4 bits, then all 77 of them at 3, which is the 96% of the 80 items quoted in the text, and none of the 64 correct should-not-call margins crosses at either width. At 2 bits the count falls back to 56 and 18 of them cross the other way, because there the margins are too small for their sign to mean much. Round-to-nearest replaces each weight with the nearest value its smaller number format can represent. Table 2 gives what the builds emit. Item 54 itself reads $+19.00$ at full precision. At 4 bits it is $+16.00$, which is 84% of what it was, and at 3 bits -8.31.

Overview rendering details. Figure 1 uses Qwen3-4B-Instruct-2507. The weight panels show magnitudes of 64 weights from one group in the released checkpoint, ordered by full-precision magnitude; the quantized values use the same order. The blue surface interpolates the original magnitudes as a visual guide, while the red steps show their rounded values. This group illustrates rounding and does not locate the cause of the preference reversal. The probability bars use the Paris request, item 54: the margin between the tool-call opening token and the text-opening token `I` changes from $+19.00$ to -8.31. Probabilities are a softmax at $T = 1$ over this pair, with $p(\mathrm{Call}) = \mathrm{sigmoid}(m)$, not over the whole vocabulary. Each panel labels its taller bar, and bars near zero are drawn at their true height. The response boxes are illustrative continuations. Actual call emission is measured separately below; emitting a call does not establish its execution.

Five checks, read from the top of the bit ladder down.

At 4 bits it reads behaviour. The model's own most likely token over the whole vocabulary is one of the two we score 78% of the time on Qwen3-4B, Qwen3-1.7B, Qwen2.5-14B and Granite-3.3-8B, pooled over the tool items. On the open-ended safety opener, a different kind and a single model, Qwen3-4B refuses in 64-token completions on 70 of the 100 items that ask to be refused at 4 bits against 73 at full precision, while its margins prefer refusal on all 100 at both widths. Where the two diverge the margin moves the slower, so refusal damage read from margins is a *lower* bound on what a user would see.

At 3 bits it reads a preference; agreement with emitted calls depends on the test set. That containment falls to 49% pooled over the same four builds and, on whether-to-call alone, to 30% across them and 1.9% on Qwen3-4B, so the margins carrying our headline result are mostly a choice between two designated continuations rather than an emitted one. Decoded freely, with no constraint to the pair, the Qwen3-4B runs reproduce the direction of the margin result: full precision emits the call on 77 of its 80 should-call items and answers in text on 68 of its 80 should-not-call ones, while 3-bit round-to-nearest emits the call on none of the 160 while converting none of the 68. Every build in Table 2 moves the way its margin row does. These decoded counts are not the margin counts and need not coincide: read from margins the same full-precision model has 64 correct should-not-call margins rather than 68, because on some items the margin prefers the call token while the text the model actually produces is not a call. The check is not confined to that model. Granite-3.3-8B is a reluctant caller to begin with, decoding a call on 16 of its own 80 should-call items at full precision against Qwen3-4B's 77, and 3-bit round-to-nearest leaves it 2, while its text answers on the should-not-call side rise from 62 to 79 of 80. The base rate differs by a factor of nearly five and the direction does not: what quantization takes is the call. Granite is also where the two instruments part company most: its full-precision margins prefer calling on 74 of the 80 while it decodes a call on 16, so on this model the margin overstates the willingness to call, and the collapse it reports is an upper bound on how many calls were there to lose.

The loss is not the model coming apart. Not calling and not working are different failures. Of the 3-bit round-to-nearest replies, 22% switch script and 23% run digits, against 4% and 1% at full precision, so that build is partly broken. The exported 2-bit GPTQ build stops the same 77 calls with neither pathology and declines in fluent English (*"I'm not able to provide a function"*), so incoherence is not what produces the collapse.

The loss is not an artifact of how the question is put. The forced-choice safety instrument reads one letter, and on every item declining is B and listed second, so a build drifting toward a letter position would look the same. Reversing both, declining first and called A, moves the flip rate by 1.5 percentage points at 4 bits and 1.0 at 3, .080 to .065 and .275 to .265 over the same 200 items.

At 2 bits it reads nothing. The fitted line is flat, the surviving margin no longer depends on the original, and the two option orders part company, .610 against .305. We treat that as erasure and the end of the measurement rather than a smaller version of the same effect, and every 2-bit number in this paper is reported under that caveat.

The full-vocabulary check, whether the model's most likely next token is one of the item's two options (above), lines up with that and adds something the two-kind version cannot say. Two of the three kinds that hold are also the ones whose margins still name the token the model would emit (*anchoring*, pooled over four builds): 100% of tool-result items and 93% of argument-fill. Which-tool is the exception and it matters: its decisions hold, at 22 of 24 builds with no flip, while its anchoring does not, at 33% against whether-to-call's 30%. So which-tool survives as a preference at 3 bits,

and the other two survive as behaviour. What survives at 3 bits is therefore not surviving only as a preference: two of the four kinds keep their anchoring, which the two-kind contrast on its own cannot show, since both of its halves have lost most of theirs by then.

A.2 THE BEHAVIOURAL CHECK OF THE MARGIN COUNTS, AS A TABLE

Table 2: The behavioural check of the margin counts: every whether-to-call item decoded freely on each build, no constraint to the token pair. Full precision emits the call on 77 of the 80 should-call items and answers in text on 68 of the 80 should-not-call ones, and the counts are taken over those: *stopped* is how many of the 77 a build no longer calls, *started* how many of the 68 it converts. At 4 bits the three 4-bit builds keep almost every call, 68, 73 and 75 of the 77. The collapse, where a build produces one, is one-directional. These are not the sign changes of Table 7: the 2-bit round-to-nearest build emits neither designated token, so it stops all 77 calls while only 56 of its 80 margins change sign.

Build	stopped	started	Build	stopped	started
GGUF q8_0	0	0	GPTQ w4	2	5
GGUF iq3_xxs	0	4	GGUF iq2_m	10	5
GGUF q4_k_m	4	1	**RTN b3**	**77**	**0**
GPTQ w3	2	11	**RTN b2**	**77**	**0**
RTN b4	9	0	**GPTQ w2**	**77**	**0**

A.3 THE FORECAST: ACCURACY, TRANSFER, AND WHERE IT FAILS

A constant-width Gaussian is also the natural guess. A margin is a sum of many weakly dependent contributions, and the spread of such a sum does not follow its mean.

The circular version of Figure 3, fitted and scored on all of a condition's items, gives 0.7 points; the held-out version is the one plotted, to avoid overstating accuracy. Per axis, the same error runs from 0.6 points on the key-value cache to 1.3 on calibrated weight quantization.

How well the forecast does. We fit (c, b, σ) to *margins*, then check Eq. 3 against a *flip rate* that played no part in the fit. Be clear about what this does not test: each condition is predicted with its own constants, so it asks whether the shape is right, not whether the constants travel. On held-out halves the median error is 1.0 percentage point over the 1,154 conditions that can be split this way, 717 of them outside the matrix (Figure 3; Appendix Table 9). The median flatters us: many conditions sit at one end or the other, where predicting and observing agree easily, and the 90th percentile error is 7.7 percentage points. On the harder middle, the 337 conditions whose observed rate is between 0.2 and 0.8, the median is 4.3 and the 90th percentile 11.7. As a per-decision probability rather than a rate, the forecast is well calibrated. Appendix A.20 gives the rate and the per-decision view.

So the margin fails as a *guarantee*, since under shrinkage no size is safe by itself, and works as a *predictor* once the shrinkage has been measured. That is what makes it cheap: run the quantized build once on the test set, and every further decision of a kind you have measured can be scored from its full-precision margin alone.

You cannot borrow our constants. Constants borrowed from another model give 1.7 to 8.4 percentage points of error at 4 bits and 9.5 to 33 at 3 bits, taking the range of mean errors across model-and-test-set pairs. A borrowed forecast is accurate exactly where nothing is damaged, and wrong exactly where damage starts. Borrowing between conditions rather than between models says the same on a far larger set: holding out each of 1,685 conditions in turn and predicting it from the median constants of the others at its bit-width costs 9.6 percentage points at 3 bits against 1.1 for its own constants, and much less where little has moved (Appendix A.20 gives the 8-bit figure). One measuring pass on your own model fixes that: fitting on half a condition's decisions and scoring the other half cuts the median error from 11.3 percentage points to 2.1, over the 75 conditions of the 135 this pass covers that can be scored at all, and helps in *every* damaged whether-to-call condition. It is not free and not universal. At 4 bits it helps in fewer than half the conditions it can be scored

on, 14 of 36, one of the eight models comes out worse, and 60 of the 135 cannot be scored at all (Appendix A.20, Figure 9).

Where the account gives way. The constant-spread clause is the part that fails first, and it fails where the margin is built from a few discrete steps rather than many small contributions. Appendix A.5 gives the exception in full and what it costs the forecast.

A.4 Estimator uncertainty and the noise floor

Conditions whose slope cannot be estimated. When the full-precision margins vary too little, we cannot reliably estimate how much they shrink. We require the slope's standard error to be at most 0.10. A fifth of the main matrix's conditions do not meet this criterion. We retain them and report their flip rates: they flip more than the rest, at a median of 0.33 against 0.11. Figure 4 compares the two groups.

Estimator uncertainty and stochastic-rounding replicates. The $\pm$ values on c are regression standard errors, given one particular quantization. Applied throughout: **differences in c below 0.04 are not interpretable.** The regressor is deterministic, so run-to-run variation exists only on the quantized side, where the replicates bound it. Its bf16 values differ in sign from an f16 reference on 3.4% of Qwen2.5 items and 0.5% elsewhere (Appendix A.6), a small errors-in-variables term that can only pull c toward zero.

A.5 The rival comparison, in full

The four accounts, defined. A0 is constant noise ($m' = m + \varepsilon$); A1 the same with a drift term ($m' = m + b + \varepsilon$); A2 margin-scaled noise ($\mathrm{sd}(\varepsilon) = s_0 + s_1|m|$); and M is Eq. 2. A0 and A1 are special cases of M; A2 is not. Multiplicative against additive is not a distinction of algebraic form, since Eq. 2 can be rewritten as an additive error $(c-1)m + b + \varepsilon$ whose size grows with the margin; what the comparison rejects in the *core kinds*, whether-to-call and the two ways of asking for a refusal, is a conditional mean fixed at the original margin, with or without a constant offset, even when residual spread is allowed to grow with the margin. At the 2-bit end a full-vocabulary check agrees that the measurement has stopped meaning much: the distribution over next tokens is much more diffuse, at a median entropy of 8.0 nats against 0.01 at full precision.

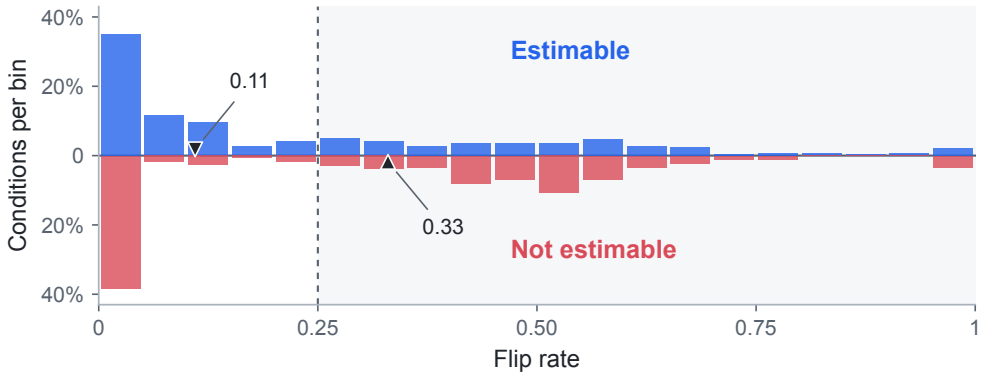

Figure 4: Conditions whose slope cannot be estimated still carry substantial damage. They make up 20% of the matrix: median flip rate 0.33 against 0.11 for the estimable conditions, with 54% losing more than a quarter of their decisions against 37%. They are reported as flip rates rather than omitted. The mirrored histograms use common 0.05-wide bins and scales, normalized separately within each group (1,039 estimable, 261 not estimable). Triangles mark medians; the shaded region marks the strict flip-rate cutoff above 0.25 used for the stated shares.

Table 3: GGUF builds (Qwen3-4B), all four quantized here from one f16 conversion: benchmark score against measured decision flips. Accuracies (%) are **BFCL v4** categories, function-calling mode throughout, scored through `llama-server`: `simple_python`, `irrelevance`, `live_simple` and `multi_turn_base`. The version matters: the benchmark's composition changes between releases, so a category score is not comparable across them. Flip shares are against the q8_0 reference, over the 226 paired decisions of all kinds and over the 106 of them on the whether-to-call side. The two flip-share columns share a numerator: every flip counted here is a whether-to-call flip and none of the other 120 decisions changes side on any of these builds, so the columns are one count over two denominators and not two measurements. The multi-turn column is scored on one machine for the top two builds and on another for the bottom two; every accuracy here is read from the scores BFCL's own scorer wrote.

Build	simple	irrel.	live-simple	multi-turn	flips (all)	flips (call)
q8_0 (ref.)	95.0	89.2	78.7	25.5	–	–
q4_k_m	95.8	89.2	81.4	24.5	.022	.047
iq3_xxs	88.0	85.8	72.5	15.0	.027	.057
iq2_m	87.5	88.3	68.2	7.5	**.097**	**.208**



Figure 5: The activation-axis grid: 60 plotted conditions, each pooled within a test set rather than fitted per kind of decision; five models, both test sets (a sixth tag repeats one model on a second machine and is held out as a replication check). Tool and safety matrices share one color scale for the surviving fraction c, with paired W8/W4 weight columns at each activation precision A16/A8/A4. White marks zero; negative values are red. At four-bit activations the weight setting stops mattering; the activation axis dominates the weight axis.

Flexible rivals rerun. A rerun with more flexible competitors added strengthens the result rather than weakening it. The competitors are an affine form with margin-growing noise, a power-curve mean, their combination, and the union of the two charitable additive rivals, an additive mean carrying both a drift and margin-scaled noise, which grants the additive account every allowance the other two grant separately. That last one wins nothing: over the 51 estimable conditions of this rerun it takes no condition at all, and the multiplicative mean beats the best additive rival in 32 of 51 whether or not it is in the pool. Over the 27 damaged conditions *no* winner has an additive mean, and once a multiplicative mean with margin-scaled noise is among the candidates, the exception kind prefers it to the additive version in every damaged condition (by 88–173 BIC), so its mean is multiplicative like everyone else's and the assumption that fails is constant variance alone. One condition deviates from linearity: the safety forced choice at 3 bits prefers a mean that bows upward (a convex power form) by 28.8 BIC over the law, a different condition from the $+28.8$ below, which is the law's *smallest* margin over the additive accounts; the other eight damaged core-kind conditions are won by the

law outright. Appendix Table 10 gives the pooled view across four quantization methods (pooling is the weaker unit of analysis, which is why the per-kind comparison above is the one we rely on); the four-account comparison, run separately on safety-refusal margins across four models and on social-bias margins, reaches the same result wherever damage is real. Model comparison of this kind selects the best predictive description among the stated alternatives; it does not prove a generative mechanism, and the competitor set, though built adversarially, is finite.

The four-account verdict, in full. The comparison is run *per kind of decision*, not pooled, because kinds of decision differ in mean margin and a pooled slope can read between-kind differences as shrinkage. Where damage is real (fitted $c < 0.70$), the multiplicative account wins every condition in the kinds the paper's claims rest on (the core kinds), and the majority within every model with three or more damaged conditions; the constant-noise and drift-only accounts win no damaged condition anywhere, per-condition BIC differences in the core kinds run $+28.8$ to $+1031.2$ (above 10 is conventionally decisive), and the verdict does not depend on the 0.70 cutoff (85.7% to 87.7% at every cutoff from 0.50 to 0.90). That choice of cutoff still selects on the fitted slope, which is the quantity separating the accounts, so the comparison is also run with no such filter at all. Over every fit, grouped by the nominal bit-width the experimenter set rather than by anything estimated, the multiplicative account wins 71.4% of 2,608, and the share tracks the ladder: 45.3% and 50.7% at 6 and 5 bits, where little is damaged and the accounts nearly coincide, against 81.1% at 3 bits and 85.9% at 2. Nor does the verdict depend on the worst-damaged settings, where the measurement is weakest: stratified by axis and bit-width the 387 core conditions are unanimous in every band, including the 50 where the margin still matches what the model would actually emit (Appendix A.22). Where quantization is near-lossless ($c \geq 0.70$) the accounts coincide and the margins between them are tiny: in the main example model's own 24 near-lossless conditions the multiplicative account wins 9, by margins of a few BIC. **One kind of decision is a genuine exception:** tool-result use prefers the margin-scaled-noise account in 89 of its 90 damaged conditions, including severely damaged ones (A2 ahead by 76 to 300 BIC, decisive in each). Of the 746 damaged conditions, the multiplicative account wins 651 and margin-scaled noise takes the remaining 95, none of them a core kind.

Where the constant-spread clause fails, measured. Residuals are normal, rank order survives at $c \approx 0$ with rank correlation up to 0.86, constant spread is rejected in 93% of tool-result and 58% of code conditions against 17% and 8% for whether-to-call and argument-fill, and weighted least squares shifts c by a median of 0.0035. Code tokens miss by 7.0 points, a third of tool-result use's 18.9.

A.6 Measurement setup in full

The four test sets cover 1,092 items: tool calling, safety refusals, general knowledge and procedure, and social bias. We built them from BFCL, MMLU, BoolQ and BBQ, and wrote our own items where no public source exists. The first axis is how we quantize the weights, and we do it four ways: round-to-nearest and an activation-aware scaling, both applied in place; calibrated GPTQ builds exported and loaded like any released model; and llama.cpp GGUF builds. Quantizing the activations as well is a second axis. Appendix A.6 sets out both.

We fit the relation separately for each run, each bit-width and each kind of decision, and separately again for each side of the three kinds that have one. There are 1,300 such rows over 822 run, kind and bit-width groups: 478 of the rows are the two sides of the 239 groups that have sides. Eighteen of the rows come from the first safety test set, which we rewrote before reading any number from it; each condition is fitted on its own items, so the two sets do not have to match, but a comparison across them is not like for like. The significance tests run over 2,088 conditions drawn from every run, which is a wider set than the matrix. The forecast is scored on a different set: 1,154 conditions that have enough readable decisions to fit on half and score on the other half, drawn from a wider set of runs. An item whose two answer options begin with the same token has no margin to read, and the forecast drops those where the matrix keeps them, which is one reason 1,300 and 1,154 differ. Appendix Table 9 reconciles every count in the paper against the runs it comes from.

One property of the fit matters enough to state here, because it decides what the tables can hold. The surviving fraction c is a regression slope, so it needs a condition's margins to vary; a kind of decision won by about the same amount everywhere has neither a slope nor an intercept we can separate, so

we report those conditions by flip rate alone. That happens in 463 of the 2,088 audited conditions, a fifth of them, and 261 of the full 1,300. They are not the quiet ones: their median flip rate is 0.33 against 0.11 for the conditions we can fit. If we dropped them we would hide the worst damage in the paper, so we keep them and report only their flip rates (Appendix Figure 4).

What is a distinct measurement, and what is held out. Three of the author-constructed kinds reuse prompts, so a row is not a trial. Counting distinct full-precision margins in the reference build, whether-to-call has 124 over 160 rows, argument filling 28 over 40, tool-result use 15 over 40, and which-tool 8 over 40: eight requests carried five times, with the same scored token pair each time, so its rows are exact duplicates and every count for that kind is over eight measurements with standard errors that the repetition understates. The safety set is 200 XSTest requests read by two instruments, 132 and 154 distinct margins over 200 rows each; the public sets (BFCL, MMLU, BoolQ, BBQ, and the seeded arithmetic) do not repeat. The forecast excludes items with zero full-precision margin, including those whose options begin with the same token; the matrix retains them. Runs that re-measure a condition on another machine are excluded from both. A separate exclusion, the depth-truncated runs, is a truncated network and a different model rather than different damage; the matrix and the rival comparison exclude it, and the held-out forecast does not, which is why 83 of its 1,154 conditions are depth-truncated builds. Held out means within a condition, fitting on one half of its items and scoring the flip rate of the other: the held-out forecast splits odd and even rows, and because that puts copies of one prompt on both sides in the three repeating kinds, §5's comparison against the fitting half's own rate is also run with rows sharing a request or a full-precision margin kept together. Request identity excludes the scored option pair. Under the request-grouped split no request straddles the halves, and on its mid-range conditions (of 861 scoreable) the law's median error is 2.5 percentage points against 4.7 for the fitting half's own rate; Appendix A.20 reports the comparison. Analyses are in sample unless stated otherwise.

Test sets, builds and the estimator, in full. What the weight runs touch. In most of these models the embedding table also turns the final state into output scores, so that table too stays at full precision and the call token's own row is never rounded; the penalty Appendix A.15 measures on that token comes from the layers before it. The coverage census counts every `nn.Linear`, tied table included; with the quantizer's own target set it reaches between 82% and 98% of the parameters across three models, and almost all of the remainder is that table, with a small additional share in non-linear-module parameters.

The four text test sets are: tool calling, 280 items (160 whether-to-call drawn from BFCL's live-simple and live-irrelevance splits, Yan et al., 2024, plus author-constructed rows for which-tool, argument filling and tool-result use, 40 each; these three kinds repeat prompts, so their items are not independent trials; Appendix A.6 gives the effective count); safety, 200 requests from XSTest (Röttger et al., 2024), each wrapped in a prompt we wrote and read by two instruments, so 400 items: 200 by forced binary choice and 200 scored on the reply's natural first token, the *safety opener*; general knowledge and procedure, 220 items (60 MMLU, 60 BoolQ, 60 synthetic arithmetic, 40 code completions); and social bias, 192 items from BBQ (Parrish et al., 2022). Appendix A.16 adds two imaged test sets and one text-only governance test set. The quantizers are round-to-nearest at group size 64 applied in place; GPTQ builds calibrated and exported with AutoRound (4, 3 and 2 bits on all 16 models); and four llama.cpp GGUF builds of each model quantized here from one f16 conversion with the importance matrix of Table 5, their margins read against that f16 conversion (the Qwen3-4B benchmark comparison of Table 3 takes `q8_0` as its reference). The two decision tables read the arithmetic kind on 300 items instead of these 60, built the same way (the answer is the first digit of the sum of two two-digit numbers); no other analysis uses these 300. The two tables also read which-tool on public tasks: 200 BFCL v3 live_multiple tasks and 200 MetaTool tasks (tool selection among the listed tools), each scored like our which-tool items, by the first token of the correct tool's name against the first token of one listed tool whose name starts differently. Argument filling is read the same way on 300 BFCL live_multiple tasks whose correct call has a string argument: the call is opened with the correct tool and parameter name, and the margin is between the first token of the correct value and of that parameter's value in another task. A condition is *estimable* when its slope carries a standard error of at most 0.10; 463 of 2,088 audited conditions are not (22.2%), and 261 of the 1,300 matrix conditions (20.1%; Appendix Fig. 4). Which conditions are unestimable depends on the kind of decision, not on chance: which-tool is 78% unestimable, whether-to-call 3%, the safety opener 0%.

What was quantized, and how. Five quantization settings, each scored on the same test sets:

- **Round-to-nearest**, in place at group size 64: 4, 3 and 2 bits on all 16 models, and 8, 6 and 5 bits on five of them; tool, safety, general and BBQ test sets.
- **GPTQ**, a calibrated export made with AutoRound: 4, 3 and 2 bits on all 16 models; the same four test sets.
- **Activation-aware scaling**, in place: 4, 3 and 2 bits on all 16 models; the same four test sets.
- **GGUF**, quantized here with llama.cpp: q8_0, q4_k_m, iq3_xxs and iq2_m on all 16 models; the same four test sets, and BFCL v4 benchmark scores on Qwen3-4B.
- **Activations**, with and without rotation: A16 to A3, 5 on the grid, 4 on the ladder, 3 rotated; tool and safety test sets.
- **Public tool tasks**: 200 BFCL live_multiple and 200 MetaTool which-tool tasks under all four weight quantizers, and 300 BFCL argument-fill tasks under round-to-nearest, activation-aware scaling and GPTQ.

The 16 weight-axis models: Qwen3 0.6B/1.7B/4B/32B; Qwen3.5 2B/4B/9B; Qwen2.5-Instruct 1.5B/7B/14B; Granite-3.3-8B; Gemma-3-4B; Gemma-4-E4B; Llama-3.1-8B and Hermes-3-Llama-3.2-3B (both Llama-derived); and Phi-4-mini. The 8-, 6- and 5-bit sweeps cover tool and safety decisions on Qwen3-1.7B, Qwen3-4B, Qwen3.5-4B, Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct; Qwen3-4B also has general-task measurements at these widths. Measured alongside but outside the 16 are four base-model controls (Qwen3.5-4B-Base, Qwen3-1.7B-Base, Qwen2.5-7B-Base, Qwen2.5-1.5B-Base) and, on the activation axis only, Qwen3-8B. The activation grid uses Qwen3-4B, Qwen3-8B, Qwen3.5-4B, Gemma-3-4B and Granite-3.3-8B; the ladder swaps Gemma-3-4B for Gemma-4-E4B, which §7 does not count, and the rotation drops Qwen3-8B.

What the GPTQ column depends on. Two things could make the GPTQ numbers in Table 1 particular to our setup, the text we calibrated on and the particular quantized model that one GPTQ run produced; a third, the 16-bit precision of the full-precision model we compare against, concerns every column. We tested each.

Changing the calibration text changes the knowledge, procedure and refusal numbers, but not whether-to-call breaking before which-tool. We took four models and quantized them a second time with GPTQ, calibrating on 256 documents of pile-10k, a public sample of the Pile (Gao et al., 2020), instead of the 61 tool-calling dialogues of Table 5 (13–18 of them per model pass AutoRound's 1,024-token minimum). Whether-to-call flipped at almost the same rate as before (within 3.1 points), and which-tool still flipped nothing. The knowledge, procedure and refusal numbers did change from their values under the dialogue calibration, by up to 0.62 in c, some up and some down.

Quantizing the same model with GPTQ a second time gives slightly different numbers but the same answer on whether-to-call breaking before which-tool. GPTQ is not deterministic: two runs on the same model give two slightly different quantized models. We ran it a second time on five models, on another machine, with the same code and calibration text, and scored both results. In all ten model-and-bit-width pairs at 4 and 3 bits the two runs agreed: which-tool flipped nothing in either, and whether-to-call flipped most among the four tool kinds in both. A single flip rate could still differ between the two runs by up to 13 points (Phi-4-mini at 3 bits: 41% in one run, 54% in the other). Each GPTQ number in the tables comes from one run.

The full-precision model we compare against is itself slightly noisy, because it runs in 16-bit floats (bf16), as the quantized models do. On Qwen2.5-1.5B, a float32 run and the f16 conversion agree in sign on 1,090 of 1,092 items, but the bf16 run differs from both on 73. Across the 16 models, bf16 differs from f16 in sign on 3.4% of Qwen2.5 items (6.7% on Qwen2.5-1.5B) and 0.5% elsewhere. For GGUF we therefore compare against the f16 conversion; the other three quantizers are compared against bf16, so this noise is in their numbers.

Token agreement along full-precision continuations. The full-precision model greedily generates sixty continuations. At every generated position we record the gap between its two highest-scoring tokens. The quantized model is then scored on the same prefixes, rather than generating its own continuation. A disagreement means its highest-scoring token differs from the full-precision token at that position. At 4 bits, 9.3% of positions disagree, and the median full-precision gap is 1.0 logits at

those positions against 14.4 where the tokens agree. The gap predicts agreement with an AUC of 0.918. At 3 bits, 38.6% of positions disagree, with an AUC of 0.816. This check uses the top two vocabulary tokens along a fixed trajectory; it is separate from the designated-token measurement and does not measure divergence between two independently generated continuations.

A.7 LIMITATIONS IN FULL

How much of each model the quantizer reaches. Counting every `nn.Linear` with the quantizer's own target set, it reaches 81.9% of the parameters of Qwen3-1.7B, 90.3% of Qwen3-4B and 97.5% of Granite-3.3-8B. Almost all of what is left is the embedding table, which every run keeps at full precision, so a build we call 4-bit is above 4 bits in effective terms.

A fifth of the full matrix cannot be fitted and is reported as flip rates instead. Appendix A.4 defines the estimation criterion and shows that these conditions have a higher median flip rate, so this limits what we can model, not what we measured.

Multiple comparisons are corrected in one place and not the other. Benjamini–Hochberg covers the per-condition tests of $c = 1$ and $b = 0$ across 2,088 conditions, not the tallies quoted downstream.

One split cannot be measured at all. Refusal against compliance is unmeasurable on GGUF builds, whose interface returns log probabilities for only the most likely next tokens, which covers under half our items and skews toward close decisions. We report it as unmeasured.

Where the measurement stops meaning anything. A full-vocabulary check locates the boundary. At 4 bits the model's most likely next token over the whole vocabulary is one of the item's two options in 78% of tool items, pooled over four builds, so the margin still reflects what the model would generate. At 3 bits that share falls to 49%, and at 2 bits to zero on every build. The pooled figure hides two splits. Across kinds, at 3 bits tool-result items are still at 100% and argument-fill at 93%, while whether-to-call is at 30% and tool selection at 33%. Across models, the pooled 3-bit share runs from 30% on Qwen3-4B to 66% on Qwen2.5-14B, so where the instrument stops reading behaviour is the model's own. So at 3 bits both halves of the dissociation in §4.1 are read mostly as a preference between two designated continuations rather than as free generation. That is what grammar-constrained decoding executes. The same check restricts the activation results further, and marks 93 of the 387 damaged core conditions to be read the way the 3- and 2-bit weight rows are (Appendix A.17).

A.8 THE EDGES OF THE ACCOUNT

Three things the account does not reach.

We cannot say why it happens. We measured several internal quantities and report them as correlates only, because no experiment here changes a computation and makes the margin move. One explanation we can rule out is that quantization partly undoes instruction tuning: the offset b is already there in models that were never instruction tuned (Appendix A.18).

We could not tell whether it holds outside text. We built the same measurement for Qwen3-VL-4B working from a screenshot, on two test sets: a *grounding* set and an *act-or-not* set. Each had to need its screenshot, or a vision result is a language result in disguise, and each had to keep a margin matching what the model would emit. The grounding set passed both checks. The act-or-not set, the one that would have answered our question, failed both, and failed them on the side that carries the result (Appendix A.16). **We therefore do not claim** that the one-directional collapse of §4.1 transfers to the second modality; we report not evidence against the split but a test set that cannot answer it. What the grounding set does support is narrower: the form of Eq. 2 replicates there, on both of the two damaged conditions it can be fitted on.

A test set can lie to you, and ours did. The last set is text, given to three of the text models: 240 items asking whether an agent acts without a human watching and whether a grant reaches production, balanced so that answering the same word every time scores exactly chance. Eight bits costs almost nothing. At 3 bits four of the six conditions stop reading the item altogether and answer one word to almost every one of the 120, scoring chance while looking decisive. **We do not claim** that these

models under-report unattended agents: asking the same question with its polarity reversed reverses the asymmetry, and the three models then disagree about which way it points. Ours did, and we would not have caught it without the reversed half (Appendix A.16).

The refusal ladder, in full. The same measurement applied to refusals shows a failure mode an accuracy score does not separate out. At 4 bits the standard current-generation instruct builds of 4B and above keep their refusal behaviour whichever way we ask, including the exported calibrated build (refuse side 0.000 on both). At 3 bits, counting the same way, the forced choice loses everything on Qwen3-0.6B, 93% on Qwen3-1.7B and 83% on Qwen3.5-2B, against 37% on Qwen3.5-9B, 20% on Qwen2.5-7B-Instruct, 14% on Qwen3.5-4B, 13% on Qwen3-4B, 2% on Granite-3.3-8B and none on Qwen2.5-14B-Instruct; the 9B is the exception to the ordering by size. Under stochastic rounding rather than the deterministic rule, the same Qwen3-4B build loses 87 to 100% across five replicates, so at this width the rounding rule matters as much as the model. Back at 4 bits, outside the set that holds, counting a loss as a refusal made at full precision and not made after quantization, over the refusals made at full precision: Qwen2.5-1.5B base loses 43% of them, Qwen3.5-2B 14% and Qwen2.5-7B-Instruct 7%; inside it, a no-zero-point rounding variant of Qwen3-4B loses 5%. Qwen3-1.7B-Base makes too few refusals at full precision to score, and loses all 13 of them. One base model acquires 17 refusals it did not make at full precision and loses none, which is why gains are counted separately. Within one kind size orders the amount lost at 3 bits on the more sensitive of the two: 0.93 on the 1.7B, 0.13 on the 4B, 0.00 on the 32B. That last leg is measured on the first safety set, the only round-to-nearest run we have for that checkpoint; a calibrated build of it on the current set agrees at zero. Across the whole roster size does not order the amount, as the 9B above shows. The observation in Appendix A.18 that model size does not simply explain the asymmetry concerns its *direction*, so the two statements are consistent. At 2 bits roughly half the refusals are gone in every model but one (Phi-4-mini flips only 6–16% of its safety decisions there, a robustness outlier we note rather than explain), and in one case the two instruments disagree qualitatively (Granite: 0.03 forced-choice against 0.64 natural-opener), which we report rather than resolve. A free-generation check validates the measurement against behaviour. On the natural-opener items, 64-token greedy completions deliver explicit refusals at an unchanged rate at 4 bits (70% against 73% at full precision), at 49% at 3 bits while the first-token margins still prefer refusal (the responses that stop refusing deflect the request rather than comply), and at 2 bits the completions are incoherent text that neither refuses nor complies. Where margin and behaviour diverge, the margin is the slower to change: refusal damage read from margins is a lower bound on the behavioural loss, not an upper bound.

A.9 An overall score does not tell you where to spend the bits

Three published observations sit at three points on one curve, each at a bit-width where c has not yet fallen far: flat benchmark scores beside accumulating agent failures, and refusal directions and single-turn tool use both surviving 4 bits. The three-build comparison in Appendix A.12 gives the same ranking by benchmark score change and by margin flip rate.

Neither whether-to-call nor refusal damage can be read off a benchmark score, because averaging over decisions that flip sharply looks smooth (Appendix Table 3).

The mismatch has a sharper form on a vision–language agent, where the behavioural rate is directly observable rather than inferred: there a language benchmark registers the damage and still points at the wrong layers. This result needs no margin at all, since it is measured on what the model emits. A deployer checks a quantized build on perplexity, and on a whole-model build that check does work: 3-bit group-64 quantization multiplies perplexity by 2.26 while the share of items producing a parseable tool call falls from 0.910 to 0.065. No whole-model build we ran keeps perplexity flat while its calling collapses, but perplexity still cannot say which layers to protect. We quantized nine neighbouring four-block slices of Qwen3-VL-4B's 36-block decoder, one at a time. Four of them hurt tool calling, with Wilson intervals clear of the baseline; the worst three cost between 0.5 and 3.4% of perplexity and remove a quarter to a half of the calls. Blocks 24–27 sit between two of them and cost nothing at all. The slice with the highest perplexity cost, blocks 0–3, is only the mildest of the four that hurt calling. On that model, perplexity therefore gave no guidance on which layers to protect. Nine slices are nine uncorrected comparisons, and the claim rests on the size of the gaps rather than on their individual p-values.

Which layers cost the score, and which cost the calls. Table 4 quantizes four contiguous blocks of the 36-block decoder to 3 bits and leaves the rest at full precision, one slice at a time. Perplexity is worst when the first four blocks are quantized, at 1.21 times the full-precision value, and the tool-call rate there is .800 against a full-precision .910. The slice that costs the most calls is blocks 16 to 19, which halves the rate to .450 while leaving perplexity at 1.01, within a point of untouched. A score computed over free text and a count of decisions that changed rank the layers differently, which is the failure of aggregation this section describes at the level of whole builds, seen one slice at a time inside a single model.

Table 4: Perplexity and tool-call rate under the same 3-bit slice interventions.

blocks	base	0–3	4–7	8–11	12–15	16–19	20–23	24–27	28–31	32–35
perplexity (×fp16)	1.00	**1.21**	1.08	1.02	1.02	1.01	1.01	1.01	1.03	1.13
call rate	.910	.800	.945	.870	.975	**.450**	**.665**	.910	**.640**	.875
95% CI lo	.862	.739	.904	.816	.943	.383	.597	.862	.571	.822
95% CI hi	.942	.850	.969	.910	.989	.519	.727	.942	.703	.914

A.10 THE REPAIR LEDGER

Each group names the baseline its entries are measured against: one more bit at a matched byte budget, the same bytes spent at random, or no byte-matched run at all. Numeric entries are median shares over the conditions that intervention applies to; the others say what was found instead.

Intervention	Flips undone
Against one more bit, at the same byte cost	
Per-item offset, oracle labels	0.260
One more bit (the baseline)	**0.305**
Per-item offset, one global estimate	0.090
Per-item offset, per kind on a calibration split	0.000
Against the same bytes spent at random	
"Important" weights, three definitions	no better than the same bytes spread evenly
Full-precision trailing blocks	no better
Margin-gradient rounding	no better
Dithering	no better
Attention-only protection	better in 3 of 7 models
Ruled out without a byte-matched run	
Layer order by margin sensitivity	its own reverse wins 6 of 16
Post-hoc logit rescaling	0, by algebra

A.11 RELATED WORK

Decision-level evaluation is not new. Dutta et al. (2024) show answers flipping both ways while accuracy holds; Dong et al. (2026) measure the same top-two gap and report it shrinking at 4 bits, and Bellam & Kim (2025) report the logit-scale attenuation qualitatively; Kiselev (2026) give the decision geometry at classifier scale. None gives a rate: how fast the gap closes per bit, which kinds close first, and a forecast that holds out of sample. Chang et al. (2025) ask which inputs break and answer in the weights, we answer with the margin. GPTQ, AWQ, rotation and the drop below 4 bits are well documented (Frantar et al., 2022; Lin et al., 2024; Ashkboos et al., 2024; Ouyang et al., 2025); precision-aware scaling laws (Kumar et al., 2025) predict a bit-width's overall *loss*, ours which decisions carry it. A concurrent agent study (Jang et al., 2026) explains quantized tool-calling failures as noise flipping the decisions whose margin is already thin: we agree on where and correct how (§3). Chhabra & Khalili (2025) find the refusal direction preserved; we measure the refusal margin instead, and it holds at 4 bits and not below, a loss an adversary can trigger (Egashira et al., 2024). Salfati (2026) state the threshold argument at one bit-width; we add the range of widths. Asadi et al. (2026) argue that multimodal benchmarks are often answerable without the image. We ran their check on both of ours before quantizing anything, and one of the two failed it (Appendix A.16). Closest is Xu et al. (2026). They lose refusals under KV-cache quantization at unchanged perplexity, measure flips per prompt as we do per decision, and find collapse onsets spanning four bits with no

universal safe width. Attention-based protection is inconclusive on our own measurement (§4.3). We differ on mechanism: they place the cause in a low-dimensional activation subspace. We find that importance-based protection does no better than the same byte budget spent at random (§4.3). Our cache runs give their account something to predict: a quantized cache damages decisions selectively by kind, not uniformly.

Where our claim is not the new part. Proskurina et al. (2024) report that quantization disproportionately affects samples the full-precision model was least confident about. That is margin selectivity, published in 2024, and no phrasing in this paper should suggest we found it. What we add is a rate and a functional form for it. Rababah et al. (2026) propose an agreement metric between base and quantized predictions that is independent of accuracy and run it from 8 down to 2 bits, reporting non-linear breakpoints; the decision-level move is theirs as much as ours, and the parts we can claim are the fitted attenuation, the split between kinds of decision, and the held-out forecast.

Where the literature disagrees with us. Jin et al. (2025) find that quantizers which ignore the massive values in self-attention lose contextual-knowledge performance, while quantizers aware of them do not, which makes any claim about "quantization" rather than about one rounding scheme conditional on the algorithm. Our matrix carries an activation-aware quantizer alongside round-to-nearest for exactly this reason, and the fitted form holds in both. Zhou et al. (2026) fit task-stratified scaling laws and conclude that reasoning is the precision-critical skill. Their axis is a task taxonomy across benchmarks and ours is kinds of decision inside tool use, so the two orderings are not comparable, and we do not test theirs: our test sets contain no multi-step reasoning items that would let us.

The three retrodicted observations, in full. Before anything new is measured, margin shrinkage explains three published observations that were reported separately. Benchmark scores staying flat while agents accumulate failures (Jang et al., 2026) is the aggregate projection of margins shrinking beneath per-item decision thresholds; Table 3 compares benchmark scores with measured flip shares. Refusal directions surviving 4-bit quantization (Chhabra & Khalili, 2025) is the law at bit-widths above the collapse: essentially no refusal flips at 4 bits in the current-generation instruct models of 4B and above, with c near 1 on both of our safety instruments in three of the four models measured on both, and the same instruments lose half their refusals at 2 bits, which a robustness account stated without a bit-width axis cannot predict. And single-turn tool use surviving 4 bits in agentic benchmarks (Dong et al., 2025) is the whether-to-call collapse sitting one bit below the operating point those benchmarks tested. One law, three prior observations, no per-observation story. The bridge is measured: on the three GGUF builds where we have both the benchmark score and the margin measurement, the benchmark's deltas and the measured share of flipped decisions order identically (the live function-calling benchmark of Yan et al., 2024; benchmark $+2.7$, -6.2, -10.5 points against flipped shares .02, .03, .10).

A.12 Benchmark scores against the margin instrument

Table 3 expands the three-build comparison above into the full grid: one model's four GGUF builds, all quantized here from one f16 conversion, scored on four categories of the live function-calling benchmark (Yan et al., 2024) beside the margin instrument's flip shares on the same builds. The single-turn categories fall by up to 10.5 points without announcing a collapse, and irrelevance detection ends within a point of where it started (89.2 to 88.3, dipping to 85.8); the multi-turn category falls from 25.5 to 7.5; and the margin instrument sees the flip mass grow $4.4\times$ over the same span, on both denominators. Every score is read in function-calling mode. Categories scored on two machines are mixed here: where a build was scored on both the single-turn categories agree to 0.5pp, but no build has a multi-turn score on both, so the column carrying the largest drop is the one with no cross-machine check. Table 5 documents the custom calibration set behind the calibrated quantization paths.

Table 5: The custom calibration set used for the calibrated quantization paths. General text and tool-calling dialogues are mixed at roughly 60/40 by tokens; dialogues are rendered through the target model's chat template so the calibration stream contains the real tool-call special tokens. The two consumers take different cuts of it: the llama.cpp imatrix path takes the 60/40 mixture, truncated at 2048 tokens, while the AutoRound export path takes the rendered dialogues alone at 1024. To be released with the code.

Component	Source	Rendering
General text (∼60%)	bartowski `calibration_datav3` (multilingual, code, technical; the community-standard imatrix text)	raw text
Tool dialogues (∼40%)	Team-ACE ToolACE and NousResearch hermes-function-calling-v1, 300 dialogues sampled, 61 surviving rendering and the length filter	chat template, real special tokens
Consumers: the AutoRound export path (GPTQ and AWQ builds; the 61 rendered dialogues only) and the llama.cpp imatrix quantizations (all 1,143 mixed documents).		

A.13 THE ROTATION INTERVENTION

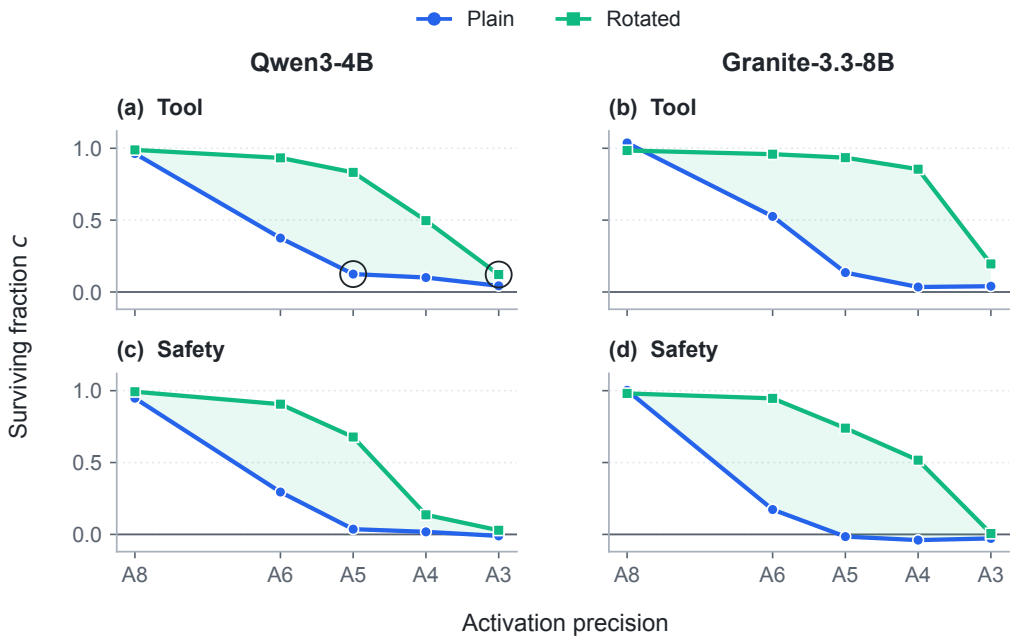

Figure 6: The same activation ladder under naive per-token min–max and block-Hadamard rotation (Plain and Rotated), with weights held at 8 bits. Columns identify models and rows identify test sets; shading separates the measured trajectories, with true bit-width spacing preserved. Outlined points mark the matched plain A5 and rotated A3 tool conditions. In both, the should-call side flips .963 of its decisions against .200 on the should-not-call side, to three decimals apiece; those are flip rates, not values of c, and the agreement is forced by saturation rather than measured, as the saturation analysis below explains. The range compression of the worst tensor predicts the shift with a mean absolute error of 0.20 bits on the two models it was developed on and 0.51 bits over all six model-and-test-set conditions.

What the rotation is worth per model. At 8-bit weights and 4-bit activations into the linear layers, the plain scheme leaves the whether-to-call surviving fraction at or below 0.04 on the four models measured both ways. With the rotation it is 0.11 on Qwen3-4B, 0.21 on Granite-3.3-8B and 0.56

on Qwen3.5-4B, against 1.00 with the activations unquantized; the forced-choice refusal behaves the same way, 0.01, 0.25 and 0.63. The fourth model measured both ways, Gemma-4-E4B, reaches 0.77 on both instruments and is the best result the rotation gives; it is also the model §7 leaves out of the counts on this axis, because it keeps 35.5% of its parameters at full precision, so it is reported here and not counted in the three above. The rotation is therefore worth about three bits of activation width on average and much less than that on some models, which is why the collapse shift and not a restored margin is what we claim.

Weights under 4-bit activations. Weight precision barely matters once the activations are at 4 bits: on whether-to-call across the five models, dropping the weights from 8 bits to 4 moves the surviving fraction by at most 0.051 and the flip rate by at most 8 percentage points, against flip rates already between 50% and 84%.

What the rotation does *not* do is remove the collapse, and that is the part that matters here (Figure 6). What it does not change is the *kind* of collapse it reaches: both schemes end one-directional toward inaction rather than symmetric, which is not automatic, since round-to-nearest at 2 bits flips both sides (Table 7). We claim no more than that from this pair. The two conditions matched here, plain 5-bit and rotated 3-bit activations, both send every whether-to-call margin to the not-call side, so their identical flip rates and flipped items are forced by that saturation rather than measured; and the $c = 0.12$ they share is the pooled slope over the whole tool test set, which §3 says not to use.

Further appendix material (the full result matrix; survival curves across bit-widths; the base-against-instruct comparison inside the collapse region; the map from injected noise to margin loss; controls that ablate the directional offset; and the repair tests with their cost-matched controls) will be released with the code.

Rotation bit-widths. Those bit-widths are properties of a quantization scheme, and we identify which scheme by measuring an alternative one. The activation quantizer above is per-token min–max with no outlier handling, the setting a substantial line of systems work is designed to escape (Ashkboos et al., 2024). Re-running the same runs with a block-diagonal Hadamard rotation around the quantizer moves the collapse **about three bits lower**: the plain scheme fails between 8-bit and 6-bit activations and the rotated one between 4 and 3, though what survives at 4 bits differs by model, from 0.11 on Qwen3-4B to 0.77 on Gemma-4-E4B, and in all six model×test-set conditions over three models the 6-bit setting is rescued outright (0.17–0.57 to 0.91–1.03). Interpolating where the surviving fraction crosses 0.70 puts the shift at 2.8 bits. **The shift is derivable in advance, but less precisely than two models suggested.** A per-token min–max quantizer's step is the token's range over 2^B, and an orthogonal rotation preserves the norm while shrinking the range, so a range compression of R is worth exactly $\log_2 R$ bits, measurable from forward passes with nothing fitted. On the two models we measured first it works: 2.24 predicted against 2.50 observed and 2.18 against 2.14 (Qwen3-4B, tool and safety), 2.46 against 2.91 and 2.40 against 2.45 (Granite-3.3-8B), a mean absolute error of 0.20 bits. A third model does not follow. On Qwen3.5-4B the predictor gives 1.88 against an observed 2.96 and 2.00 against 3.17, missing low on both test sets and taking the mean over six conditions to **0.51 bits**. That model's decoder-block linear targets reach 84.9% of its parameters, with the embedding table and small non-linear modules left at full precision, so coverage alone does not establish comparability; it uses linear attention where the other two use standard attention, a difference the $\log_2 R$ argument does not account for and which we have not tested. We therefore claim the sign and the order of magnitude, roughly two bits, enough to plan a bit budget against, and not the precise shift. The predictor must be the 90th-percentile input to the MLP down-projection (`down_proj`), not the median tensor (which would predict 0.96 bits): the collapse is set by the worst-conditioned tensor. What the rotation does to that tensor is exactly what "spreading the outliers" should mean; its excess kurtosis falls from 141 to 3.

A.14 The repair grid

The law says where help must come from, and the two choices that move the surviving fraction are its own suggestions. *What if the bits were spent where the law says?* Moving bits from the weights to the **KV cache** (the attention keys and values stored and reused during generation) inverts the usual allocation, and a weight×cache grid on two models and both test sets says how far. An 8-bit cache is free: over 30 checks, two models, five kinds of decision, three weight settings, the largest change in c is 0.037, below the 0.04 we treat as noise. A 4-bit cache is not, and its damage is selective in

the same way the weight axis is, costing a median 0.30 in c on the forced-choice safety test and whether-to-call against 0.08 on the safety opener, argument filling and tool-result use; 3 bits takes the first two to zero. The ladder barely interacts with weight precision, the same shape appearing at 16-, 4- and 3-bit weights as on the activation axis. Four bits spent on the weights leave c at 0.83–1.03 across both models' core kinds; the same four spent on the cache leave 0.57–0.81. Four models at 8-bit weights extend the axis past the grid's two models: Gemma-3-4B, Granite-3.3-8B, Qwen3-4B and Qwen3.5-4B, 18 conditions, and repeats the selectivity at a 3-bit cache: median c 0.09 on the core kinds against 0.73 elsewhere. It also shows what a single collapse point hides. The core-kind slope there spans the full range by model, -0.01 on Granite-3.3-8B and -0.02 on Qwen3-4B against $+0.83$ on Qwen3.5-4B, so no cache width below 8 bits is safe for every model: Xu et al. (2026)'s central observation, reproduced on our instrument and on their axis. One caveat applies to every cache number: the collapse point belongs to the quantizer and not to the bit-width alone. Our two cache quantizers differ only in scheme, per-token symmetric against per-token asymmetric min–max, and they agree at an 8-bit cache (10 matched conditions, largest difference 0.025 in c) while differing by as much as 0.471 at 4 and 3 bits (22 conditions). The two-model grids above are the symmetric scheme's and the four-model series is the asymmetric one's, rebuilt from the stored margins; a cache result that does not name its scheme is not reproducible. *What if the outliers were spread before quantizing?* The rotation of §4.2 is that intervention, and it buys about three bits; the range-based prediction has a mean absolute error of 0.51 bits across six model-and-test-set conditions, with larger errors on Qwen3.5-4B. A third help is narrower: a proof-of-concept activation-aware quantizer (our own implementation, not AWQ, exponent searched on a synthetic probe) helps exactly at the collapse and nowhere else (15 of 19 matched conditions at 3 bits, chance-level at 4 and 2), evidence about this implementation rather than the method class, though the two agree where they can be compared: on Qwen2.5-7B-Instruct at 4 bits a released AWQ build changes the sign of 6 of the 160 whether-to-call margins and none of the 80 on the should-call side, against 10 and 1 for ours; and non-uniform bit allocation across layers helps while the sensitivity *ranking* justifying it does not, ordering by sensitivity performing within noise of its reverse.

The exclusions, in full. Against these, the exclusions. One class is ruled out **by algebra, and it is worth being clear that this is not an experiment**: any post-hoc transform $\ell \to \alpha\ell + \beta$ applying one $\alpha > 0$ and one β to every logit scales each margin without moving its sign, so temperature scaling and a global bias offset cannot recover a single flipped decision. The scope is that kind and no wider. A correction with a per-class or per-token term, a low-rank output correction, layerwise reconstruction against the full-precision activations, or anything acting on hidden states rather than logits is not covered by the argument and is not tested here; we exclude the transforms whose failure follows from their form, not the idea of post-hoc correction. **Four fail when measured**, in the specific forms below and at a matched byte budget: protecting "important" weights (three definitions, weight norm, input-activation norm and measured margin sensitivity, each protecting nine of Qwen3-4B's 36 layers at a matched 3.25-bit average against the same bytes spread evenly across the layers: on whether-to-call, weight norm and activation norm leave c at 0.16 and 0.16 with flip rates of 0.58 and 0.56, the evenly spread control sits at 0.24 and 0.46, and margin sensitivity raises c to 0.44 while its flip rate stays at 0.43, with the anti-margin control at 0.23 and 0.49), keeping trailing blocks at full precision, margin-gradient rounding, and dithering (noise added before rounding). A fifth, attention-only protection at matched cost, wins in only three of the seven models tested and is therefore too unstable to claim in either direction. The three are Qwen2.5-7B, Qwen3-4B and Granite-3.3-8B, by median gaps in c of 0.131, 0.072 and 0.133 over their shared kinds; the four losses include Llama-3.1-8B at $+0.024$, smaller than the 0.04 we treat as noise. **One requires knowing each item's correct answer (an oracle), and its label-free variant only diagnoses**: subtracting the per-item measured offset takes a median 0.260 off the flip rate when told which answer is correct, over the 61 conditions where it applies. Estimating b per kind on a calibration split and subtracting it, which needs no labels, recovers flips exactly where the law says damage is bias-dominated (35 of 46 on the largest one-directional condition, net $+0.65$) and nets **0.0 at the median** over the damaged conditions, because where decorrelation dominates there is no bias to remove. The practical results are two: the three importance definitions we tried consistently fail against cost-matched controls, which is evidence against those definitions rather than against importance-aware quantization as a research direction. Every entry is measured against the same baseline: **one more bit takes a median 0.305 off the flip rate** over the 501 severely damaged conditions, which is the drop in the share of decisions that change sign and not the share of the flips it undoes; every repair above is measured the same way, so the column is comparable. Nothing tested *without ground-truth labels* beats it.

A.15 THE MEASURED CORRELATES

What the offset is made of. The margin sees b through two tokens, so a b fitted on one pair cannot separate a drift that moves the whole vocabulary from a penalty on the call token. Recording the change in every vocabulary logit separates them (160 whether-to-call items per model). Both are present and only one of them matters. There is a real vocabulary-wide drift, and its direction is not even shared: at 3 bits the median logit moves -7.7 on Qwen3-4B, -2.4 on Granite-3.3-8B and $+0.9$ on Qwen3-1.7B. What the three agree on is the call token, which lands at the 0.0th, 0.3rd and 0.5th percentile of that distribution, below almost every other token in the vocabulary, while the alternative it is scored against sits at the 88.7th on Granite. So the offset in Eq. 2 is not the vocabulary shifting under the pair. It is the call token being singled out, which is why the direction of the collapse is a property of the decision rather than of the model's overall drift.

And not of the token's kind. The call token is the only one in the test set that is reserved rather than ordinary, so a quantizer that treats rare reserved tokens harshly would put it at the bottom for reasons that have nothing to do with tool calling. Measuring the class alongside it settles which is happening. Two classes are recorded per item: every other reserved token in the vocabulary, and the ordinary tokens whose unembedding row norm is nearest the call token's, which controls for the call token merely having an unusual output row. At 3 bits the class effect exists and is not about being reserved: on Qwen3-4B other reserved tokens fall -12.9 against a vocabulary median of -7.7, and norm-matched ordinary tokens fall -12.5, the same amount. On Qwen3-1.7B neither class falls at all, sitting at the 40th and 38th percentile of a distribution whose median moves up. On no model does the class reach the call token, which is -2.8, -13.1 and -8.7 logits below its own class on Granite-3.3-8B, Qwen3-4B and Qwen3-1.7B and at the 0.3rd, 0.0th and 0.5th percentile against the class's 19th, 8th and 40th. The two effects sized against each other: the norm-matched class falls 1.4, 4.8 and 1.1 logits below the vocabulary median on those three models and the call token a further 2.6, 13.5 and 8.4 below *that* class, so the token-specific penalty is 1.8 to 7.6 times the one norm explains. The triple above is measured against the reserved class and this one against the norm-matched class, which is why the two do not agree to the tenth. It is the larger of the two, and not by a fixed factor.

The repair the offset account predicts, and why it fails. If quantization pushes one token down by a constant, adding that constant back at inference should undo it, at one float per model and bit-width and no extra bits, and it can be applied after a build has shipped. Swept over the 45 damaged whether-to-call conditions that still have a correct should-not-call item left to lose, the bias that maximises correct decisions is a median 6.5 logits, and it recovers a median 100% of the lost calls at the cost of a median 100% of the should-not-call items the quantized build still had right; a bias that costs the decline side nothing recovers anything at all in 13 of the 45, at a median of zero. The two sides score the same pair in opposite senses, and by the width where the penalty matters they sit on the same side of zero, so no scalar separates them. This is the additive counterpart of the algebraic result in the introduction: rescaling the logits moves no sign, and biasing the one token that is singled out moves every sign at once.

A.16 THE SECOND MODALITY, AND WHY IT GIVES NO RESULT

We built the same instrument for Qwen3-VL-4B working from a screenshot, on a *grounding* set and an *act-or-not* set, and report it here because how it failed is reusable even though what it found is not.

Two checks a modality test set has to pass. A vision result that survives without the image is a language result wearing a screenshot, and a margin that no longer tracks what the model would emit is not a decision. The grounding set passes both: without the screenshot the model prefers the target on 179 of 378 items, chance, at a median margin of exactly 0.00, and its with-image and without-image choices agree on 47.6%, also chance. The act-or-not set, the one that would have answered our question, fails both: without the image the model still prefers the target on 269 of 400 items and agrees with itself 91.8% of the time, and at 3 bits its argmax over the whole vocabulary is one of the two we compare on 38.5% of items, one-sided at 15.0% of should-act items, the side carrying the result, against 62.0% on the other.

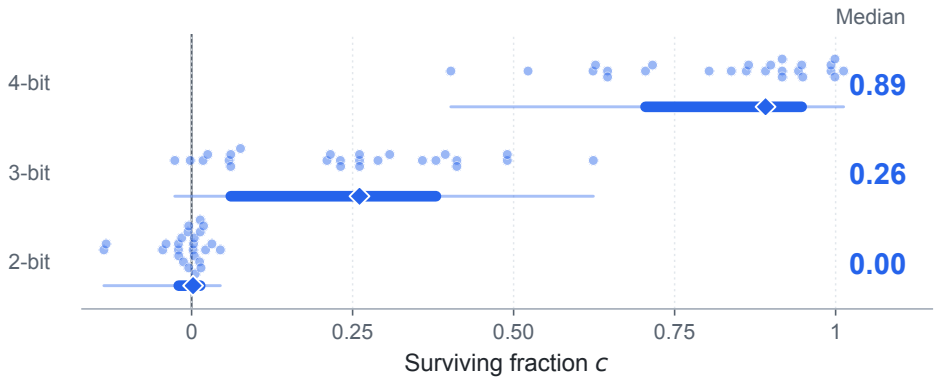

Figure 7: The whether-to-call surviving fraction of the 25 fits over 11 models that carry all three weight widths and name their model in the results to be released. Each fit contributes one point per width; vertical offsets separate overlapping points. Diamonds mark medians, thick intervals the middle half, and thin intervals the full range.

What that rules out. This set cannot say whether the one-directional collapse of §4.1 transfers. The containment failure is one-sided and aligned with the result, so the instrument cannot separate the asymmetry from its own blind spot. The same check run on text is side-symmetric (§A.1), which is why the text claim stands and this one does not. What survives is narrower and stated as such: the form of Eq. 2 replicates on the grounding set, on every damaged condition we can fit.

What is not established. Every model here is Qwen3-VL, since surveying for vision plus a chat template that accepts tool definitions returns only this one kind, so this is a single-kind result. The per-item behavioural forecast of §5 is not established here. And the 3-bit rows use the same round-to-nearest simulator as the rest of the paper, which is harsher at 3 bits than a shipping mixed-precision build: our own served `Q8_0`, `Q4_K_M` and `Q3_K_M` builds all click within three pixels of one another and all emit a valid tool call where the simulator parses none.

A.17 How far down the margin still anchors behaviour

The full-vocabulary check of Appendix A.7 run on the activation axis. Naive per-token activation quantization destroys the anchoring far earlier than weight rounding: 4-bit activations retain 1–3% of each test set's full-precision share of items whose most likely token over the whole vocabulary is one of the two designated options, against 94–97% for 4-bit weights and 34% for 3-bit weights. The rotation brings the anchoring back to 80% at 4-bit activations and not one bit further down, so it recovers the margin's behavioural meaning over the same range where it recovers any of the margin (§4.2), which on two of the four models is most of it and on the other two very little. The 93 affected conditions are those at 5-bit activations and below without the rotation, or at 3-bit activations with it, out of 387 damaged core conditions.

A.18 The one-directional collapse, by condition

Where each model falls on the way down. The collapse is universal and its timing is not. At 4 bits the fits run from 0.40 to 1.01, the values above 1 examined later in this appendix, with a median of 0.89; at 3, the width where the damage begins, they scatter widest, from 0.03 below zero to 0.62; and at 2 bits every one is under 0.05 with a median of 0.00.

Whether-to-call and which-tool (§4.1) are read from different items; the two checks below rule out the items as the cause of the gap. Within a typical condition the eight distinct selections are decided by nearly the same margin (median $|m|$ 14.8 logits, within-run spread 0.24 logits, across 145 runs), so the kind carries no difficulty gradient that could produce an item effect; and the same selections do collapse one bit lower, so the items are not immune. The 160 whether-to-call items and the eight

Table 6: Table 1 at 4, 3 and 2 bits and the four GGUF builds. Each cell is c/flip: the median surviving fraction c over the models where it can be estimated, and the median share of decisions that change sign (%), over the 16 models. A superscript gives how many models were tested on that decision when fewer than 16 were; – means that fewer than four models give a c with standard error at most 0.10. GGUF is read against the f16 conversion. Arithmetic uses 300 items. BFCL and MetaTool rows are public tasks.

	Round-to-nearest			Activation-aware			GPTQ			GGUF			
Decision	4	3	2	4	3	2	4	3	2	q8_0	q4_k_m	iq3_xxs	iq2_m
Tool use													
Whether to call	0.86/10	0.22/41	0.00/55	0.85/6	0.34/39	-0.01/50	0.85/7	0.55/21	0.06/49	1.00/1	0.92/3	0.71/10	0.53/23
Which tool, ours	–/0	–/0	–/61	0.66/0	–/0	0.02/52	0.62/0	–/0	–/0	0.99/0	0.86/0	–/0	–/0
Which tool, BFCL	0.85/1	0.25/11	0.03/48	$0.86/1^{15}$	$0.46/5^{15}$	$0.01/51^{15}$	$0.85/1^{12}$	$0.59/2^{12}$	$0.18/12^{11}$	$1.00/0^{14}$	$0.96/0^{14}$	$0.70/0^{14}$	$0.63/3^{14}$
Which tool, MetaTool	0.88/4	0.24/16	0.00/51	$0.92/3^{15}$	$0.48/11^{15}$	$0.00/48^{15}$	$0.90/3^{12}$	$0.63/6^{12}$	$0.21/14^{11}$	$1.00/0^{14}$	$0.97/2^{14}$	$0.76/5^{14}$	$0.65/6^{14}$
Argument fill, ours	0.92/0	0.61/0	0.19/51	0.96/0	0.89/0	–/47	1.00/0	0.85/0	0.59/0	1.00/0	1.02/0	0.90/0	0.83/0
Argument fill, BFCL	$0.90/1^{15}$	$0.59/4^{15}$	$0.02/50^{15}$	$0.93/0^{15}$	$0.67/2^{15}$	$0.03/49^{15}$	$0.92/1^{5}$	$0.71/4^{6}$	$0.25/34^{6}$				
Tool-result use	$0.91/0^{15}$	$0.38/0^{15}$	$0.00/28^{15}$	$0.87/0^{7}$	$0.42/0^{7}$	$0.00/32^{7}$	$0.93/0^{11}$	$0.77/0^{11}$	$0.31/8^{11}$	$1.00/0^{11}$	$0.99/0^{11}$	$0.91/0^{11}$	$0.80/0^{11}$
Refusal													
Forced choice	0.83/7	0.25/35	0.01/49	0.84/4	0.35/23	0.00/53	0.85/6	0.55/14	0.01/62	1.00/0	0.98/4	0.76/6	0.54/10
Open-ended opener	0.89/2	0.40/7	0.01/44	0.91/1	0.54/3	0.00/33	0.94/2	0.70/4	0.09/6	1.00/0	0.97/0	0.77/2	0.76/2
Knowledge													
MMLU	0.83/12	0.19/35	-0.02/55	0.91/12	0.37/32	-0.02/49	0.83/16	0.54/27	0.00/51	0.99/0	0.96/8	0.73/18	0.57/19
BoolQ	0.89/7	0.49/23	0.00/51	0.93/7	0.50/21	-0.01/52	0.91/7	0.63/12	0.11/38	1.00/0	0.95/2	0.87/6	0.75/11
Procedure													
Arithmetic	0.77/3	0.37/22	0.00/49	0.77/2	0.39/6	0.01/48	0.70/1	0.63/5	0.23/48	1.00/0	0.95/0	0.59/1	0.48/2
Code token	0.99/5	0.83/10	–/40	0.99/0	0.88/4	–/39	0.98/0	0.91/10	0.08/30	1.00/0	0.99/0	0.99/10	0.94/10
Social bias													
BBQ	0.76/10	0.26/40	-0.01/52	0.88/8	0.41/32	-0.01/50	0.84/8	0.52/22	0.04/48	1.00/1	0.93/5	0.76/15	0.58/20

Table 7: Which builds flip the decision to call a tool, and in which direction. Below the rule, the other five models at the width where the damage appears; all six at all three widths are Appendix Table 8. The two columns are the items whose correct response is a tool call and those whose correct response is a text reply, 80 items each (58 and 48 on the GGUF builds, whose interface scores only the likeliest tokens and whose bit-widths are nominal; 77 and 79 on the two Qwen3.5 models); the ratio divides the tool-call column by the text-reply column. A sign change in either direction counts, so the second column includes items the full-precision model had wrong and the quantized one gets right. † marks the builds that send every whether-to-call margin to one side: their ratios are forced by that saturation rather than measured, and the §4.1 claim about direction counts only the 101 conditions that did not saturate.

Model	Quantizer	Bits	Tool call (%)	Text reply (%)	Ratio
Qwen3-4B	Round-to-nearest	4	5.0	7.5	0.7
Qwen3-4B	Round-to-nearest	3	96.3†	20.0†	4.8
Qwen3-4B	Round-to-nearest	2	70.0	40.0	1.8
Qwen3-4B	GPTQ (exported)	4	2.5	8.8	0.3
Qwen3-4B	GPTQ (exported)	3	2.5	13.8	0.2
Qwen3-4B	GPTQ (exported)	2	96.3†	20.0†	4.8
Qwen3-4B	GGUF q4_k_m	4	3.4	6.2	0.5
Qwen3-4B	GGUF iq3_xxs	3	1.7	10.4	0.2
Qwen3-4B	GGUF iq2_m	2	13.8	29.2	0.5
Granite-3.3-8B	Round-to-nearest	3	88.8	42.5	2.1
Qwen2.5-14B	Round-to-nearest	3	2.5	25.0	0.1
Qwen3-0.6B	Round-to-nearest	3	32.5	66.3	0.5
Qwen3.5-2B	Round-to-nearest	3	80.5	44.3	1.8
Qwen3.5-9B	Round-to-nearest	3	33.8	36.7	0.9

selections are drawn from different sources but sit in one test set and share the system prompt and tool format.

What the instrument can and cannot see at 3 bits. The 3-bit whether-to-call collapse does fix a direction: of the runs where every decision has already gone the same way, the overwhelming majority go toward not calling. At this bit-width the flips are read as conditional preferences, not as emitted tokens (§A.1). So we ran here the check of Appendix A.16: the share of items whose

Table 8: Every model that carries both sides of the whether-to-call decision under plain round-to-nearest, at all three weight widths. Columns as in Table 7: the share of margins that change sign among the items whose correct response is a tool call and among those whose correct response is a text reply, 80 of each except on the two Qwen3.5 builds, which carry 77 and 79. † marks the widths that send every whether-to-call margin to one side, where the ratio is forced by that saturation rather than measured. The leaning is not shared: at 3 bits three of the six flip the tool-call side more than the other and three flip it less, which is the spread the direction tally of §4.1 counts over its larger population.

		Margins flipped (%)		
Model	Bits	Tool call	Text reply	Ratio
Qwen3-4B	4	5.0	7.5	0.7
	3	96.3†	20.0†	4.8
	2	70.0	40.0	1.8
Qwen2.5-14B	4	1.3	6.3	0.2
	3	2.5	25.0	0.1
	2	48.8	42.5	1.1
Qwen3-0.6B	4	35.0	27.5	1.3
	3	32.5	66.3	0.5
	2	82.5	45.0	1.8
Qwen3.5-2B	4	6.5	32.9	0.2
	3	80.5	44.3	1.8
	2	37.7	46.8	0.8
Qwen3.5-9B	4	5.2	8.9	0.6
	3	33.8	36.7	0.9
	2	59.7	60.8	1.0
Granite-3.3-8B	4	5.0	16.3	0.3
	3	88.8	42.5	2.1
	2	92.5†	46.3†	2.0

full-vocabulary argmax stays inside the option pair at 3 bits is 30.6% on the should-call side and 28.8% on the other, over four builds, against 77.2% and 46.3% at 4 bits (§A.1). What is nearly even is where the two sides end up: 30.6 against 28.8. What is not even is how far each fell to get there, 46.6 percentage points against 17.5. The asymmetry needs only the even end points, not even falls. At the bit-width where the 4.8-fold result is read, the instrument is equally blind on both sides, so its blindness cannot make one side look worse than the other.

The checks on the split, in full. The two halves come from different items, 160 whether-to-call against eight distinct selections carried as 40 rows, so the split could in principle be an item effect. Two checks say it is not. Within any one condition the eight selections have nearly the same margin, so no difficulty gradient could produce a split. The same 40 do collapse one bit lower, so the items are not immune: the kind of decision sets the bit-width of the collapse. A third check swaps which kind has margin to spare, on the same model and quantizer: ten which-tool items whose two tools are near-synonyms, median margin 6.2 logits, against ten whether-to-call items whose request is impossible without the tool, median 22.4. If the split were distance from crossing, the small margins would flip and the large ones would hold. At 3 bits the large ones flip 10 of 10 and the small ones 5 of 10: giving whether-to-call more room does not save it and taking which-tool's away does not finish it. What differs is the drop itself: -35.2 logits for whether-to-call against -7.1 for which-tool on the same build, which is the kind of decision and not the item. A fourth pairs the two kinds by margin rather than trusting them to be alike. Within a condition the eight selections carry nearly the same margin, and whether-to-call is spread wide, so which-tool could survive because its items were written with margins larger than any offset the quantizer produces. Measuring both kinds at full precision first and pairing items across kinds to within 1.5 logits gives 24 pairs on Qwen3-4B at medians of 15.03 and 14.94. At 3 bits, on those pairs, whether-to-call changes side on 17 of 24 and which-tool on none. The pairing is what limits this: on Qwen3-1.7B only 6 pairs exist and on Granite-3.3-8B none do, because there the two kinds' margins do not overlap at all, at medians of

3.75 and 13.38. That the two kinds differ in margin is exactly the objection, and it is answerable only where they overlap.

A fifth check varies the token the decision is read against. Both sides of the kind score the call token against `I`, so the headline is measured through one alternative on all 160 items, and what collapses could in principle be that token. Rescoring the same prompts against `The` and against `Sure` leaves the collapse where it was: on Qwen3-4B the should-call side changes side on 77 of its 80 items at 3 bits against `I`, on 77 against `The` and on all 80 against `Sure`, and on 4, 2 and 2 of them one bit higher. The bit-width of the collapse does not move with the alternative on any model tried. The size of the collapse does: on Granite-3.3-8B the same count is 71 of 80 against `I` and 53 of 80 against `The`, so the instrument reads the decision while how much of it moves depends on what the decision is read against.

What the tool-result kind could and could not read. The tool-result rate is read over 25 of that kind's 40 items. In the other 15 the two temperatures on offer share a leading digit, and these tokenizers split a number into digits, so the two scored tokens are one token, the margin is identically zero and the item cannot change sign whatever quantization does. No item in the other three kinds is like that, and counting the 15 in or out leaves every 3-bit median and zero-flip count above unchanged. A second defect is that Qwen3.5's chat template will not render the tool-result turn these items carry, so the kind cannot be measured on those builds at all. Both are repaired in a separate test set that moves the fifteen distractors off the leading digit and stores the tool-call arguments as a mapping rather than a string, which is the form that template requires and which renders byte-identically on the builds that already worked. Re-measured there, all 40 items are live on all four builds tried, two of them previously unmeasurable, and the flip rate is zero at 4 bits and at 3 on every one, against a median 0.51 at 2. What survives at 3 bits is the kind, not the items that happened to be readable.

Table 9: Where each denominator comes from. "Estimable" means the slope's standard error is at most 0.10; "damaged" means a fitted $c < 0.70$. Each row lists its unit, source and filter. Each analysis uses a different set of conditions, so a count from one row does not apply to another.

count	unit	source and filter
1300 / 1039	matrix rows / estimable	main matrix: 93 runs × kind × bit-width, intercept estimator; the 261 unestimable rows are the 20% described in Appendix A.4
2,088 / 463	audit conditions / unestimable	the per-condition tests of $c = 1$ and $b = 0$ over every run; 22.2% report only a flip rate
1,625	forecast-computable conditions	full-condition fits with se$(c) \leq 0.10$ and residual $\sigma > 0$; the held-out forecast is a separate population, half-fit, zero-margin items dropped, 12 items a half
1617	audited conditions	constant-spread audit: wider set of runs, $n \geq 24$, estimable
1,685	leave-one-out transfer conditions	each predicted from the others' median constants at its bit-width
27	damaged comparison conditions	per-kind model comparison, main example model's conditions, fitted $c < 0.70$
746 / 1997	damaged / all per-kind conditions	the same comparison over every run
117 / 101 / 68	direction conditions / unsaturated / leaning toward not calling	the direction tally over every whether-to-call run: at least 20 paired items per side and a flip rate above 0.20 on at least one side; 16 send every margin to one side and are dropped
17	models in the direction tally	the same 117 conditions counted by model, including base controls and the activation-only Qwen3-8B
46	offset-law conditions	whether-to-call conditions of the main matrix that carry both sides
24	four-kind builds	3-bit builds carrying all four tool kinds in one run
65 / 20	per-model fits / 3-bit pairs with both halves	weight axis, one row per model, quantizer and bit-width that carries whether-to-call
91	other-kind conditions at 3 bits	the three tool kinds other than whether-to-call, for the overlap count
1,154	held-out conditions	Figure 3: odd/even split within each condition, all damage axes; a wider set of runs than the matrix, 717 of them outside it, and items with no readable margin dropped
748	repair-test conditions	calibrated-offset repair: damaged conditions with at least 5 flips
501	repair-baseline conditions	the one-more-bit baseline: matrix conditions with flip ≥ 0.25
1,575	clustered-closure conditions	the by-model clustering of Appendix A.21: per-condition closure recomputed over every run, five unattributable runs excluded
1,605 / 203,160	calibration conditions / predictions	per-item flip probabilities: conditions with $n \geq 24$ and se$(c) \leq 0.10$

Table 10: Does quantization add noise or shrink the signal? Pooled fits per condition; the per-kind comparison is in §3. The first four rows are Qwen3-4B and the last two name their own model. ΔBIC is the law's margin over the best of the three additive accounts, and > 10 is decisive; a negative entry means one of them fits better. Four entries are negative: on the two deeper GGUF builds noise growing with the margin wins, while GGUF q4 and GPTQ w4 are undamaged (c of 1.00 and 0.98), where the accounts coincide and the margin is within noise. The correlation column is evidence against constant-noise accounts only; R^2 is fit quality only. RTN is round-to-nearest.

Condition	survival c	fit R^2	ΔBIC (shrinkage)	corr($\lvert m\rvert$, $\lvert m' - m\rvert$)
RTN b4 / b3 / b2 (tool)	0.90 / 0.46 / 0.02	.95 / .51 / .01	+56 / +236 / +508	.02 / .24 / .94
RTN b4 / b3 / b2 (general)	0.90 / 0.30 / −0.03	.88 / .14 / .01	+12 / +75 / +481	.12 / .57 / .90
GGUF q4 / iq3 / iq2 (tool)	1.00 / 0.81 / 0.59	.99 / .95 / .79	−5 / −81 / −57	−.11 / .47 / .71
GPTQ w4 / w3 / w2 (tool+general)	0.98 / 0.79 / 0.37	.96 / .88 / .18	−3 / +168 / +211	−.05 / .32 / .40
RTN b4 / b3 / b2 (tool, Qwen3-0.6B)	0.81 / 0.05 / −0.04	.70 / .04 / .02	+27 / +531 / +487	−.15 / .81 / .86
RTN b4 / b3 / b2 (tool, Llama-3.1-8B)	0.78 / 0.85 / 0.09	.93 / .75 / .09	+200 / +21 / +629	.64 / .27 / .94

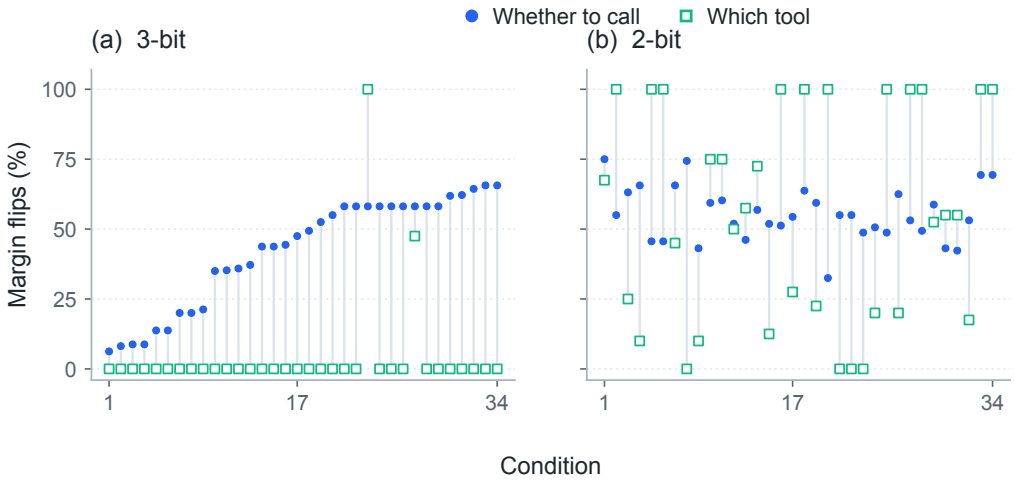

Figure 8: The invocation/selection split across the same 34 model-and-method conditions. Within each column a connector pairs whether-to-call with which-tool, preserving exact zero rates. The 3-bit and 2-bit panels use the same condition order, sorted by the 3-bit whether-to-call rate. Which-tool is untouched in 32 of the 34 conditions at 3 bits. At 2 bits, which-tool is untouched in only 4 of 34. Zero flips do not establish a zero population rate, and the forty which-tool rows are eight requests carried five times (Appendix A.6).

Controls on the split. **The direction tally.** The tally itself selects 117 whether-to-call conditions with a flip rate above 20% on at least one side, a filter on observed flips rather than on the fitted slope. Of these, 16 have every margin on one side after quantization; such a condition flips whatever the full-precision model got right rather than what quantization did, so they are dropped, and fifteen of the 16 dropped conditions saturate toward not calling and one toward calling. Of the remaining 101, 68 lean toward not calling, at median flip rates of 0.68 on the should-call side against 0.37 on the other ($p = 0.0006$, exact two-sided binomial).

The direction, counted by model. Counted over the 17 models represented in this tally, including base-model controls and the activation-only Qwen3-8B, with each model's median gap taken first, 11 of 17 lean toward not calling, 5 lean the other way and 1 is exactly level ($p = 0.33$ two-sided, 0.21 without the level model). The directional tendency is therefore not statistically established at the model unit.

The offset and the lean. Over the 46 whether-to-call conditions of the main matrix that carry both sides, the median ratio of the two sides' flip rates is 0.75 below $|b|/\sigma = 1$; the ratio is 4.81 above 3, with a rank correlation of 0.22 across the whole range.

The four kinds, counted. On the 24 builds carrying all four kinds, whether-to-call changes side on every one at a median rate of 0.58, while argument filling and tool-result use show no flip at all on 16 and 18 of the 24 against which-tool's 22. The per-model table covers 65 conditions over 15 models and three quantizers at every bit-width; its 3-bit rows with both halves are 20 pairs over 13 models and three quantizers, and whether-to-call flips more in 19 of them at whether-to-call rates from 0.08 to 0.66. Two runs record neither a model nor a quantizer and are left out of it. Of the 91 conditions in the other three kinds at 3 bits, 10 reach whether-to-call's tenth percentile of 0.09.

Three things make this more than a curiosity of one condition. It is **not an artifact of margin size**. Within bands of the $|m|$ range shared by the two sides (should-call and should-not-call), permuting the side labels gives an exact null with margin size removed by construction; a directional flip-rate gap survives that test in 8 of 10 models at 3 bits (5 toward not calling, 3 toward calling) and 3 of 10 at 2 bits. In all 11 directional cases the gap's direction matches the sign of the model's fitted offset, which was fitted on margins and never on flips. Nor is the pooled line an artifact of the two margin clusters, though the two sides sit in separate ranges of m and a single line could in principle be fixed by the two cluster centres alone: where the pooled slope reports deep damage ($c < 0.30$;

121 side-labelled conditions), the larger of the two within-side slopes has a median of 0.10 and exceeds 0.5 in 1 of the 121 conditions. In the flagship condition the two within-side slopes refit to 0.24 and 0.10, with the same large negative offset. A small pooled c is therefore not hiding an intact side. It **survives on an exported build**: the exported GPTQ w2 model collapses one-directionally too, while above the collapse (w3, $c = 0.81$) there is no asymmetry yet, the law's own prediction. And what it tracks is **the offset relative to the residual noise**, not the method. Rank the builds of Table 7 by $|b|/\sigma$, which the table itself does not carry: the two at 3.8 and above are the two that flip one-directionally, Granite at 2.2 sits at an intermediate $2.1\times$, and every symmetric row is at 0.8 or below, including RTN b2, whose c has also reached zero but whose offset has not. (One slope sits above 1: GPTQ w4, at 1.03 ± 0.02. Mild amplification at shallow damage is a matrix-wide pattern, not a fluke of this row: 14% of estimable conditions at 4 bits and above sit above 1, with median excess 0.045, under plain rounding and calibrated methods alike. The excess here is within the 0.04 floor of Appendix A.4.) The asymmetry belongs to the early stage of damage, where the offset b outweighs the residual noise; at 2 bits it disappears ($c \approx 0$, flips .700 against .400, both directions). The pattern is not simply explained by model size, and we report the counts rather than force a story. Qwen3.5-9B shows no asymmetry at either bit-width, and the sub-2B models are direction-unstable, one reversing at 3 bits and showing the forward asymmetry at 2 while the other does the opposite.

Base-build control on the offset. A tempting account of the offset is reversal: quantization partially undoes instruction tuning, so the model drifts back toward its pretrained prior, which would explain why the drift so often points away from acting. Matched base builds test it, because a model that never received instruction tuning has nothing to undo: the account predicts a near-zero offset, as a fraction of the model's own typical margin, in every base. The four matched pairs at 3 bits on whether-to-call show otherwise. One base's offset is four times the size of its sibling's and of opposite sign (base -0.84 against instruct $+0.20$; Qwen2.5-7B); one is comparable to its sibling's (-0.55 against -0.59; Qwen2.5-1.5B); one is near zero, as the account predicts ($+0.04$ against -0.34; Qwen3.5-4B); on one the offset cannot be fitted (Qwen3-1.7B). An offset already present in two of the three bases where it can be fitted cannot be an artifact of undoing tuning; whether reversal contributes in some models stays open.

A.19 Both halves of a call on the same external tasks

Design. Four hundred BFCL v4 live-multiple requests with one ground-truth call and at least two tools, the correct tool's position shuffled per task, 80 tasks for calibration and 320 sealed, split by task id. Each task is decoded three ways on every build, greedy, through the model's own chat template: with its tools, where the model should call the correct one; with the correct tool removed, where it should not call; and after the call opener alone, where the complete name is matched against the tool list. The same tasks carry the whether-to-call pair on both variants and the which-tool pair after the shared prefix of the names, so margins and emitted decisions are read on one item set. Three templates render the same prompt with and without the tools argument: Gemma-3, Hermes-3 and Phi-4-mini receive the definitions as JSON in the system turn instead, and each model's pair uses the token it emits, a fenced block for Gemma and a brace for the other two. A build enters only if at full precision it opens and names a call on at least 0.8 of the tasks when nothing forces it to, and names the correct tool on at least 0.8 when the call format is given to it. Granite-3.3-8B calls unforced on 0.66 and Phi-4-mini on 0.71, announcing the tool in prose on the rest, so both stay out.

The two tests, fixed before the tasks were built. Among test tasks correct under both full-precision instruments, invocation damage must exceed selection damage by at least 15 points on at least three builds, and the gap averaged equally over the builds must have a 95% lower bound above 10 points when whole tasks are resampled; round-to-nearest and calibrated runs are reported apart, and a claim about deployment needs the calibrated runs to repeat the result. For prediction, the law must beat the calibration split's prevalence by at least 10% in Brier score on emitted failures, with a paired interval excluding zero. Both are read from the stored runs.

How the calls are lost. The three builds that lose the call first lose it in different ways, and the way matters for what a deployer sees. Gemma-3-4B and Qwen3-4B stop opening it, on 80% and 57% of the tasks they opened it on at full precision; Qwen2.5-7B opens it every time. Gemma-3-4B writes prose that names the tool and never opens the call, "Okay, I will use the `Hotels_2_SearchHouse`

model	3-bit run	calls lost	names lost	gap (pp)	95% lower
Gemma-3-4B	round-to-nearest	0.87	0.32	+55.3	+48.7
Qwen2.5-7B	round-to-nearest	0.57	0.12	+44.3	+37.7
Qwen3-4B	round-to-nearest	0.75	0.55	+19.7	+12.5
Qwen3-1.7B	round-to-nearest	0.93	0.96	−3.5	−7.4
Hermes-3-Llama-3.2-3B	round-to-nearest	0.44	0.59	−15.0	−22.7
all five, equally weighted				+20.2	+16.9
Qwen2.5-7B	AutoRound-GPTQ	0.18	0.02	+15.7	+11.0
Qwen3-4B	AutoRound-GPTQ	0.00	0.01	−1.0	−2.6

Table 11: Invocation and selection losses on the same subset of the 320 sealed tasks: those correct under both full-precision instruments. Each rate uses that shared denominator, which is 260 tasks on Hermes-3, 285 on Qwen3-1.7B and 300 on the other models; gaps use unrounded rates. A call counts when it is opened and names a tool, in the model's own format. The test fixed in advance asks for a gap of at least 15 points on three builds and a lower bound above 10 on the five-build average; the round-to-nearest runs give both.

…", which is a model declining to act. Qwen3-4B opens a call on 42% of the test tasks against 97% at full precision, and finishes a parseable one on 24%, so most of its loss is the decision and the rest is the format. Qwen2.5-7B keeps opening the call on 97% of them and finishes a parseable name on 43%, so its failure is the format breaking around a decision that held. The calibrated Qwen2.5-7B build fails the same way one step milder: of its 53 lost calls, 41 emit the call token and then the end-of-turn token immediately, and its margins put the whether-to-call sign change at 0.006. Both calibrated builds also call more often when the correct tool has been removed, 0.31 against 0.16 on Qwen3-4B and 0.29 against 0.23 on Qwen2.5-7B, so calibration moves the error from silence toward over-calling.

Margins on the same tasks. On Qwen3-4B at 3 bits the two halves keep nearly the same share, $c = 0.12$ for whether-to-call against 0.24 for which-tool, and separate in their offsets, -14.6 logits against -0.6. Gemma-3-4B repeats that shape at $c = 0.10$ against 0.19 with offsets -3.6 and $+0.5$. Qwen2.5-7B at 3 bits holds both, $c = 0.49$ and 0.51, which is why its lost calls are formation rather than decision. Hermes-3-Llama-3.2-3B is in the emitted results and not here: at full precision only 0.43 of its should-invoke margins are positive on tasks it calls on 81% of the time, so on the prompt its template forces the pair is close to a tie and reads nothing.

Prediction. Forecasting each test task's emitted invocation failure from its should-invoke margin, with (c, b, σ) fitted on the calibration tasks, beats the prevalence forecast at 4 bits on Qwen3-4B by 38% in Brier score (95% CI 25 to 51) and fails everywhere else. At 3 bits it fails in both directions: on Qwen2.5-7B the margins keep their sign and 57% of calls fail to form, and on Qwen3-4B nearly every margin changes sign while a quarter of the calls still come out. The forecast test is therefore not met, and §5 states the forecast over margin sign changes, which is what it is fitted and validated on.

A.20 The forecast, in full

Against the calibration half's own rate. A deployer who has run the calibration half already holds that half's observed flip rate, so the forecast has to beat it to be worth fitting. Scored on the same held-out halves and the same 1,154 conditions as the forecast above, the law's median error is 1.0 percentage points against 1.3 for the fitting half's own rate, 7.7 against 9.0 at the 90th percentile, and the law is closer on 46% of conditions with 20% tied, the ties being conditions where nothing flips on either half. On the 337 conditions whose observed rate lies between 0.2 and 0.8 the gap opens: 4.3 against 6.0 at the median, the law closer on 66%, and an item-level Brier score of 0.185 against 0.239. Argument filling and which-tool carry each request five times, so a split that cuts between rows puts one request on both sides; splitting instead by the request as posed, with the scored option pair excluded from its identity, keeps the law ahead of the fitting half's rate and widens the middle-range gap to 2.5 against 4.7. That split covers the 861 conditions whose rows can be traced back to the request that produced them, and no request straddles it; the rule it replaced, which grouped rows by their full-precision margin, leaked a request across 269 of those same conditions.

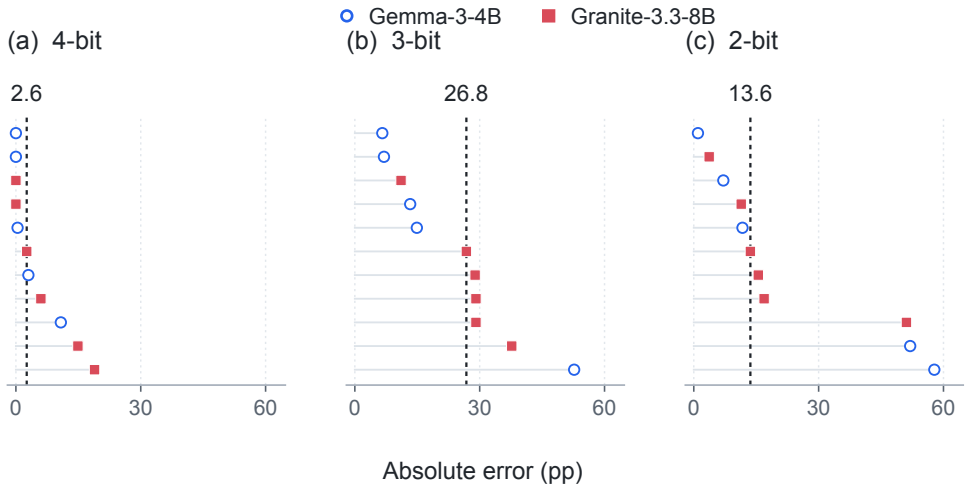

Figure 9: Borrowed constants, applied to one model held out of the matrix and one weaker control: Gemma-3-4B is genuinely held out; Granite-3.3-8B contributes elsewhere in the matrix. Eleven forecasts per bit-width are ranked by absolute error; marker shape identifies the model and each labelled dashed line marks the median. At 4 bits the median error is 2.6 points and at 3 bits it is 26.8, where a deployer has to measure their own model rather than borrow ours.

The calibration pass, condition by condition. The limits appear when constants transfer across models, and we measured them. Applying another model's fitted constants to the two transfer-test models gives 1.7–8.4pp of error at 4 bits and 9.5–33pp at 3 bits; one of the two (Gemma-3-4B) is genuinely held out, while the other (Granite-3.3-8B) contributes elsewhere in the result matrix and is the weaker test. The forecast is accurate where nothing happens and wrong where damage begins (Figure 9). One calibration pass per model repairs exactly the case that needs it: fitting (c, b, σ) on half a condition's decisions and scoring the other half cuts the median error from 11.3 percentage points to 2.1 over 75 scoreable conditions on eight models, and the gain is not uniform. At 4 bits it helps in fewer than half the conditions, 14 of 36, because constants borrowed from another model already work where quantization has broken nothing, the count rather than the medians, which are not stable there (3.0 to 3.0 over all eight models, 3.0 to 2.1 with Gemma-4-E4B dropped, against a count of 14 of 36 and 14 of 33); at 3 bits it takes 23.2 points to 2.1 and helps in 18 of 20 conditions, and at 2 bits 16.9 to 5.6. On whether-to-call it helps in *every* damaged condition, 17.0 to 0.3 at 3 bits and 30.4 to 0.7 at 2. Seven of the eight models improve on their own median and Gemma-4-E4B does not, going from 5.3 points to 8.5. Sixty of the 135 conditions cannot be scored because the slope is not identifiable on half the data, so the pass is worth running and is not a substitute for the measured table; extrapolating from a single measured bit-width is worse than that table (mean absolute error 36.5 against 14.9pp). Borrowing across conditions rather than across models says the same thing on a much larger set of fits. Holding out one whole condition at a time from the 1,685 estimable ones and predicting its flip rate from the median constants of every other condition at its bit-width, the median absolute error is 9.6 points at 3 bits against 1.1 for the condition's own constants; widening the donors to the whole matrix gives 11.3 over those same 1,685. At 8 bits the same borrowing costs 1.6 points, because little has moved for the constants to be wrong about. So the constants are a property of the condition at the bit-width where the damage is, which is the case for running the pass rather than reading ours.

Forecast accuracy, in full. The median is flattered by conditions whose observed rate already sits at or near an endpoint, and the distribution behind it is heavy-tailed. The whole tail runs 3.5 points at the 75th percentile, 7.7 at the 90th and 10.4 at the 95th; by damage band the median error is 0.4 at $c \geq 0.70$, 1.3 in 0.30–0.70, 3.4 in 0–0.30 and 3.4 below zero; and the 817 conditions whose observed rate is at or near an endpoint carry a median of 0.5, which is what flatters the headline. Read as a per-item uncertainty estimate, the formula is well calibrated: pooling 203,160 item-level predictions from 1,605 estimable conditions, each from its condition's own fit, decisions given a flip

probability of p flip at a frequency within 0.02 of p in every probability band (expected calibration error 0.004; Brier score 0.080 against 0.166 for guessing the average rate). It does not degrade as damage grows (medians 1.3, 1.9, 2.0pp at 4, 3, 2 bits). The distinction worth keeping: the margin fails as a *certificate* (under contraction no size is safe by itself) and succeeds as a *predictive feature* once the contraction is calibrated. The calibration pays off in its cost structure. The quantized build is run once, on the test set: a few hundred paired margins per model and bit-width, each costing one quantized and one full-precision forward pass to the first token. For every further decision of a kind we have calibrated, the flip probability is then estimated from its full-precision margin alone, without running the quantized build on it. Counting flips directly would cover only the test set's own items; the calibrated formula covers any decision the full-precision model can score. That is the saving over a behavioural evaluation, which runs the quantized build over every item at generation length. Conditions from the same model are not independent; Appendix A.21 reports the held-out forecast and the other summaries with each model counted once.

A.21 Condition accounting

The main *matrix* contains 1,300 conditions. Its 93 runs cover mostly weight-quantization experiments. Additional experiments test activations, rotation and the KV cache. The model comparison uses 366 runs; the held-out forecasts and activation analyses also draw on additional runs. Table 9 lists the population for each analysis. A condition is one model configuration evaluated at one bit-width on one kind of decision in a single run. Repeated runs remain separate, and flip rates use the readable items in each group.

For the three kinds whose items have a should and a should-not side, the matrix holds each condition three times: both sides together and each side alone. Dropping the pooled entries leaves the medians of §3 unchanged. Conditions from one model are not independent observations, so the law's in-sample error and the one-more-bit baseline are also reported clustered: each model's median taken first, then the median of those, which counts a model with many runs once. Checking Eq. 3 against each condition's own observed flip rate, in sample, the error moves from 1.14 points per condition over 1,575 conditions to 1.24 per model over 23 models, and the one-more-bit baseline from 0.305 over 501 conditions to 0.345 over 12 models. Both changes are small; both summaries are fitted and scored on the same items. The held-out forecast survives the same treatment: 1.4 points per model over 20 models against 1.0 per condition, on the 1,125 of its 1,154 conditions whose model is recoverable. So does the rival comparison, which is a share rather than a median: pooling each model's damaged conditions into one proportion, the multiplicative account wins a majority in all 22 models with at least five damaged conditions, with a median share of 0.92 and no model below 0.70. Five runs whose model cannot be recovered from their records are excluded rather than pooled into an unlabelled bucket.

Table 9 reconciles the counts from analyses with different scopes and filters.

A.22 The verdict by way of quantizing and bit-width

Table 12: Damaged core-kind conditions ($c < 0.70$, slope estimable) against how far the margin is anchored to behaviour there. Anchoring is the full-vocabulary argmax share retained relative to the same model and test set at full precision, since the safety test set's absolute share is low (21–24%) even undamaged. Rotated 4-bit activations keep the anchoring (79–80%) and rotated 3-bit ones do not (0–15%).

Condition	Conditions	M wins	Anchoring
Weight $\geq$4 bits	37	37	94–97%
Activation A6 tool, rotated A4	13	13	79–80%
Activation A6, safety	9	9	53%
Activation $\leq$A5, rotated A3	93	93	0–15%
Weight 3 and 2 bits	221	221	0–34%
Unresolved (A8, rot. A5, KV)	14	14	not run
Total	387	387	

The free-generation check that anchors the margin to behaviour was run on the weight axis only; the full-vocabulary check now covers the activation axis as well, and both verdicts depend on bit-width (§7). If the model comparison of §3 drew its result from the depths where those checks are weakest, the result would be an artifact of the instrument rather than a finding about quantization. It does not (Table 12): the multiplicative account takes every band unanimously, including the 50 conditions where the margin still matches what the model would actually emit: 37 at 4-bit-and-above weights, the only setting where the free-generation check was also run, and 13 in the rescued activation group (9 at rotated 4-bit activations, 4 at naive 6-bit tool), where the full-vocabulary check alone establishes it.

The verdict is unanimous in every row, so it does not come from the rows where the instrument is weakest. What the stratification does establish is where the margin may be read as behaviour: 4-bit weights keep 94–97% of the full-precision argmax share, 4-bit activations 1–3%, and the same 4-bit activations with the rotation 80%. Activation quantization is therefore not simply a second axis of the same damage, without the rotation it leaves the instrument's behavioural anchoring far earlier than weight rounding does, and the 93 conditions at 5-bit activations and below, or at 3-bit activations with the rotation, carry the same caveat as the 3- and 2-bit weight rows. The fourteen conditions of the unresolved row are held out of these bands: 8-bit activations, rotated A5, and the one that quantizes the key–value cache rather than the layer inputs. Their anchoring has not been measured, and the cache condition is a different intervention from the ones the bands were measured on.