THE BUTTERFLY EFFECT IN LARGE LANGUAGE MODEL QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantization has been widely used to reduce the memory footprint and inference latency of large language models (LLMs). Low-bit quantizers have typically been evaluated using average metrics such as perplexity and task accuracy. Nevertheless, these average metrics do not reveal failures on individual inputs, such as correct answers becoming wrong after quantization. Focusing on tail reliability, we propose a unified source-to-decision analysis, which measures rare output failures and traces each input's quantization errors from individual modules through downstream amplification to changes in output distributions, token decisions, and autoregressive generation. Our analysis reveals the quantization butterfly effect: High-gain computation paths amplify small rounding errors on particular inputs into large output shifts and token changes. These amplified errors form an upper tail hidden by average metrics, and an early token change redirects the autoregressive continuation. Our experiments demonstrate the quantization butterfly effect on Qwen2.5-3B, where 4-bit weight-only GPTQ increases mean perplexity by only 7.3%, yet the 99th-percentile (p99) per-token KL divergence is 8.9 times the mean, 14.8% of top-1 predictions flip, and almost every greedy continuation leaves the full-precision trajectory. The same pattern recurs across model families and zero-shot, reasoning, code, and instruction-following tasks. Similar average accuracy scores do not show how often correct predictions become wrong. Our perturbation–propagation theory factorizes each module's impact into local error and downstream gain along the error's direction. We prove non-asymptotic KL bounds, derive an exact propagation-aware decision radius and a sharp multiclass KL barrier for top-1 changes, and establish guarantees for preserving greedy trajectories; finite-perturbation bounds extend Fisher-based control to every quantile and CVaR. Our theory-guided diagnostic score identifies intervention targets for causal validation. Restoring only the four highest-scoring layers to full precision while keeping all other layers at 3 bits reduces the p99 divergence gap by up to 83%. To improve tail reliability, we introduce two complementary mitigation methods: Propagation-Aware Precision Allocation (PAPA) protects high-impact modules at a fixed bit budget, while Butterfly-Aware Reliability Tuning (BART) learns a lightweight correction at a fixed bit-width. By updating under 2% of parameters, BART raises 3-bit GSM8K accuracy from 0.030 to 0.350. Tail reliability must therefore be measured and reported alongside average performance, and optimized as a quantization objective.

1 INTRODUCTION

Quantization reduces the memory footprint and inference latency of large language models (LLMs), making LLMs smaller and faster to serve. Quantization is already used in practical LLM deployment, including Meta's 4-bit Llama models for mobile devices (Meta, 2024). LLM quantization is typically evaluated using average metrics, such as perplexity (PPL) and task accuracy. Advances in post-training quantization (PTQ) keep these metrics close to full-precision (FP) baselines (Dettmers et al., 2022; Frantar et al., 2023; Xiao et al., 2023; Lin et al., 2024; Shao et al., 2024; Chee et al., 2023; Ashkboos et al., 2024). A small change in these average metrics is then taken to mean that quantization has barely changed the model.

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.

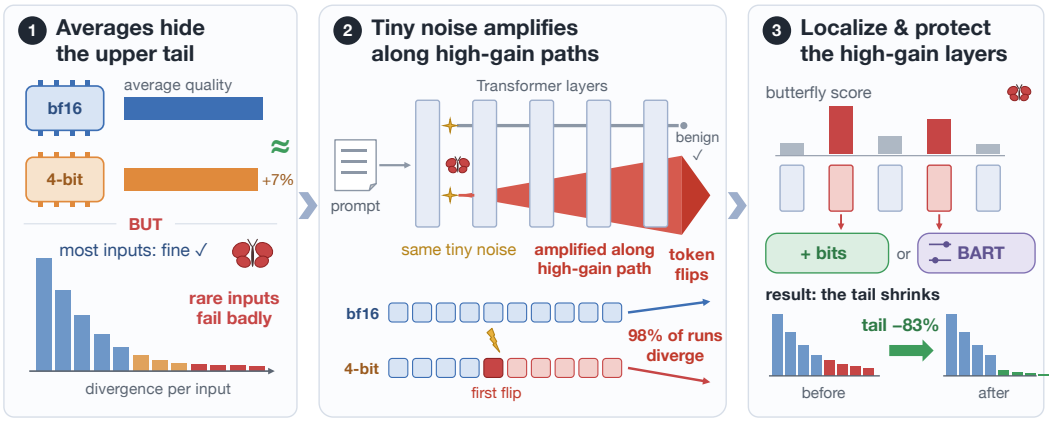


Figure 1: The quantization butterfly effect and mitigation. The schematic abbreviates bfloat16, the FP reference in this work, as bf16. (1) Average quality can remain close to FP while amplified errors on a subset of inputs form a pronounced upper tail of divergence. (2) Small local perturbations remain benign on stable paths but grow along high-gain Transformer paths; crossing a token decision boundary changes the context and branches the autoregressive trajectory. (3) Propagation-aware diagnostics identify high-impact components; selective precision protects high-impact components, while Butterfly-Aware Reliability Tuning (BART) learns a lightweight correction to reduce the upper tail. Quantitative results are reported in Secs. 3.1 and 3.2.

Although these average metrics summarize performance over many inputs, deployment succeeds or fails on the particular request in front of the user. A quantized model can preserve nearly the same average score while turning correct answers into wrong answers, making previously passing code fail the corresponding tests, or breaking instruction following that previously succeeded (Sec. 3.3). We observe that gains on some inputs offset failures on others, so stable average accuracy hides changes in which requests the model answers correctly (Sec. 3.3 and Appendix L.2). Prior work on compression has reported that aggregate accuracy conceals individual answer flips, and low-bit quantization disproportionately harms particular inputs (Dutta et al., 2024; Chang et al., 2025). In other words, quantization can leave most inputs nearly unchanged while causing large errors on a specific subset. Benchmark coverage adds a second blind spot. Large errors on types of deployment requests missing from the evaluation datasets never enter the reported averages.

We study the quantization failures hidden by average metrics: why small rounding errors produce large output changes on particular inputs and how quantization-induced output changes affect deployment. The mechanism is input-dependent error amplification. The same small local error stays benign on a stable computation path but grows along a high-gain path. Large amplified errors on a subset of inputs form the upper tail that average metrics obscure. When inputs are ordered by output-change magnitude, inputs with the largest amplified errors occupy the high-error end of the distribution, which forms the upper tail. When an error crosses a token decision boundary, the changed token becomes context for the next step, sending the rest of autoregressive generation down a different path. We call this chain from small rounding errors to amplified output changes and branching generation trajectories the quantization butterfly effect.

Our experiments demonstrate the gap between average performance and tail reliability. On Qwen2.5-3B, 4-bit weight-only GPTQ (GPTQ-W4) raises mean PPL by only 7.3%, yet the 99th-percentile (p99) Kullback–Leibler (KL) divergence between the FP and quantized next-token distributions is 8.9× the mean, and the top-1 prediction changes at more than one position in seven (Table 4). The generation-level consequence is widespread. In the larger null-controlled sweep, 98% of greedy continuations leave the FP trajectory within 48 tokens (Sec. 3.1). These early trajectory bifurcations capture the butterfly effect's central intuition, sensitive dependence on initial conditions, in autoregressive decoding (Lorenz, 1994).

Tail reliability means keeping output errors small and decisions stable even on the inputs most affected by quantization. We formulate tail reliability as a source-to-decision quantization problem. Our framework follows a realized low-precision error from the injection site, through downstream gain

along the error's direction, to a finite output-distribution change and an exact decision boundary. This source-to-decision chain makes rare failures measurable against a numerical-noise null, failure sources causally testable, and failure severity directly actionable through selective precision and lightweight tuning. Figure 1 summarizes the phenomenon, the mechanism, and the mitigations that follow from the framework.

Our perturbation–propagation theory proves bounds on output divergence and establishes conditions for preserving token decisions and greedy trajectories (Sec. 2). Specifically, quantization injects a local error at each Transformer module, and the downstream computation suppresses or amplifies that error along the error's actual direction. An error entering an amplifying path can produce a large logit change, while an equally large local error on another path remains harmless. The output impact is the local error multiplied by the directional downstream gain. We then solve the forward problem, which asks how large the resulting KL can be, and the inverse problem, which asks for the smallest source perturbation that can cross a decision boundary, and extend the link through greedy decoding.

We validate the perturbation–propagation framework experimentally and derive practical diagnostics and mitigations from the framework. Our contributions are:

- We formulate tail reliability as a source-to-decision problem. Our decomposition tracks each module's realized error on each input along the direction in which the rest of the network amplifies or suppresses that error. A non-asymptotic KL certificate, exact propagation-aware decision radius, sharp multiclass KL barrier, and greedy-trajectory certificate then connect the same source to continuous output shift and discrete behavior; finite-perturbation bounds transfer the Fisher surrogate to every quantile and conditional value at risk (CVaR) (Sec. 2 and Appendix E).
- We establish the empirical and causal signature of quantization tail risk. Controlled FP-to-FP nulls and evaluations across model families, quantizers, key–value (KV) caches, and real tasks expose a pronounced divergence tail, harmful per-item flips, and early trajectory bifurcations despite benign averages. The directional score then causally localizes the tail: Restoring four selected layers reduces the p99 divergence gap by up to 83%, outperforming local-error and curvature selection on average (Sec. 3). Together, the tail, harmful flips, and early branching expose the deployment consequences of input-dependent error amplification.
- We provide two tail-targeted mitigation methods validated across models and tasks: Propagation-Aware Precision Allocation (PAPA) and BART. PAPA protects the modules whose errors matter most under a fixed bit budget, while BART learns a lightweight correction at a fixed bit-width. Experiments demonstrate reductions in tail divergence and recovery of degraded task performance with PAPA and BART (Sec. 4).

2 Perturbation–Propagation Framework

For each input, we follow a quantization error from the Transformer module where the error arises to the model's next-token decision. Here, a decision is the token with the largest output logit; changing the selected token changes the context for later decoding and can redirect the generated response. We track the error's actual direction through the remaining network to explain both large output-distribution changes and changes in the selected token. All proofs are in Appendix F.

2.1 Propagation Model and Error Amplification

Let B_ℓ denote an FP Transformer block, x_ℓ the block's input residual state, and $z \coloneqq R(x_L)$ the FP logits. For a source perturbation $\delta_{\ell,m}$ at module m's output, define $J_\ell \coloneqq DB_\ell(x_\ell)$, $\Phi_{a:b} \coloneqq J_{b-1} \cdots J_a$, $C \coloneqq DR(x_L)$, and $J_{\ell,m}$ as the Jacobian from module m's output to block ℓ's output. The quantized-minus-FP logit change $\Delta z \coloneqq \hat{z} - z$ has the local perturbation–propagation approximation

$$\Delta z \approx \sum_{\ell,m} \underbrace{C\Phi_{\ell+1:L} J_{\ell,m}}_{\text{gain } G_{\ell,m}} \delta_{\ell,m}. \tag{1}$$

Rounding quantizers are discontinuous at thresholds, so the expansion serves as a local surrogate. The block recurrence, intermediate derivation, and remainder conditions are given in Appendix E.1.

Local fragility and downstream gain. For $\delta_{\ell,m} \neq 0$, define the directional gain of the linearized source by $\mathrm{Gain}^{\mathrm{dir}}_{\ell,m} := \|G_{\ell,m}\delta_{\ell,m}\|/\|\delta_{\ell,m}\|$ and set the directional gain to zero when $\delta_{\ell,m} = 0$. For module m in layer ℓ, the output impact is the local error magnitude multiplied by the directional downstream gain:

$$\mathrm{Impact}_{\ell,m} = \|G_{\ell,m}\delta_{\ell,m}\| = \underbrace{\|\delta_{\ell,m}\|}_{\text{fragility}} \cdot \underbrace{\mathrm{Gain}^{\mathrm{dir}}_{\ell,m}}_{\text{gain}}.$$

Small local errors can be amplified by high directional gain, and both fragility and gain depend on the input. Appendix E.4 details the four regimes and the attention and gated multilayer perceptron (MLP) mechanisms.

2.2 Output and Decision Guarantees

Theorem 1 bounds the KL divergence between the FP and quantized next-token distributions from the propagated logit error, without assuming that the error is small. The bound uses the exact logit error; using Eq. 1 to estimate the range Ω therefore requires retaining the approximation remainder.

Theorem 1 (Non-asymptotic propagated-KL certificate)**.** *At a token position, write the exact logit error as*

$$\Delta z = \sum_{a \in \mathcal{A}} v_a + r, \qquad v_a := G_a \delta_a,$$

where $a := (\ell, m)$ indexes module sources and r is the exact linearization, interaction, and readout remainder. Define the shift-invariant range seminorm $\mathrm{osc}(u) := \max_j u_j - \min_j u_j$ and

$$\Omega := \sum_{a \in \mathcal{A}} \mathrm{osc}(v_a) + \mathrm{osc}(r).$$

For $p := \mathrm{softmax}(z)$ and $q := \mathrm{softmax}(z + \Delta z)$, let $F := \mathrm{diag}(p) - pp^\top$. Then

$$\begin{aligned} \left|\mathrm{KL}(p\|q) - \tfrac{1}{2}\Delta z^\top F \Delta z\right| &\leq \frac{1}{6}\,\mathrm{osc}(\Delta z)^3 \leq \frac{1}{6}\Omega^3, \\ \mathrm{KL}(p\|q) &\leq \frac{1}{8}\,\mathrm{osc}(\Delta z)^2 \leq \frac{1}{8}\Omega^2. \end{aligned} \tag{2}$$

Adding the same offset to every logit leaves the softmax distribution unchanged, so the bound depends on differences between token-logit errors, measured by $\mathrm{osc}(\Delta z)$. The second line limits the exact KL by $\Omega^2/8$, while the first limits the absolute error of the Fisher quadratic approximation by $\Omega^3/6$. The source-wise bound can be loose because the sum of source ranges ignores cancellation between sources. Both guarantees require including the remainder r; Appendix E details the components of r.

The FP winner is the token with the largest FP logit, which FP greedy decoding selects. A token decision boundary is reached when, after perturbation, that token ties with a competitor for the largest logit. The original token then no longer has a strict lead, and the tie-breaking rule determines which token is selected. If a competitor moves strictly ahead, greedy decoding must select a different token, changing the context for subsequent generation. Theorem 4 in Appendix E.3 gives the source-error radius needed to reach the FP winner's decision boundary in the propagated linear model: For each competitor, divide the FP logit gap from the winner by the propagated gain in that pairwise logit difference, then take the smallest ratio. A small gap or large gain means that only a small source error is needed to erase the FP winner's lead. Corollary 5 takes the smallest such radius across decoding steps, giving the error radius within which the entire FP greedy trajectory is preserved.

Exact KL barrier. A top-1 change requires another token to match or exceed the FP winner's probability in the perturbed distribution. The next theorem gives the exact minimum forward KL needed to remove that winner's strict lead.

Theorem 2 (Exact KL barrier for a token change)**.** *Let $p \in \Delta^{V-1}$ have a unique maximizer c, and let $a \in \arg\max_{j \neq c} p_j$ be a runner-up. Define*

$$\kappa(p) := p_c \log \frac{2p_c}{p_c + p_a} + p_a \log \frac{2p_a}{p_c + p_a}, \tag{3}$$

using the standard continuous extension $0 \log 0 \coloneqq \lim_{u \to 0^+} u \log u = 0$ *(Cover & Thomas, 2006, Sec. 2.1). For every* $q \in \Delta^{V-1}$ *under which* c *is not the unique maximizer,*

$$D_{\mathrm{KL}}(p\|q) \geq \kappa(p). \tag{4}$$

The bound is sharp: Equality is attained by $q_c = q_a = (p_c + p_a)/2$ *and* $q_j = p_j$ *for* $j \notin \{c, a\}$*. Thus* $\kappa(p)$ *is the exact minimum for loss of the unique winner and the infimum for a strict change of winner, and* $D_{\mathrm{KL}}(p\|q) < \kappa(p)$ *certifies preservation of* c*.*

The cheapest way to remove the FP winner's strict lead is to pool the winner's probability with the runner-up's probability and split the pooled mass equally, leaving every other token's probability unchanged. This boundary distribution gives $\kappa(p)$, so a measured KL below $\kappa(p)$ guarantees that the next token stays the same. A KL at or above $\kappa(p)$ does not force a token change: The direction of the distribution shift still matters. The bound covers the full vocabulary, including cases where the new winner was not the FP runner-up.

If $D_t < \kappa(p_t)$ at every FP prefix through step H, the quantized model emits the same first H greedy tokens. This stepwise KL condition turns the teacher-forced KL we already measure into a guarantee for the generated continuation. Corollary 6 in Appendix E.5 also bounds the trajectory risk by $\mathbb{E}[\sum_{t=1}^{H} D_t/\kappa_t]$, without assuming independent steps. Appendix L.2 applies the stepwise KL token-preservation test to a real chain-of-thought (CoT) bifurcation (Figure 7).

Fisher control of the divergence magnitude. The Fisher quadratic estimates the output-distribution change from the logit perturbation. Theorem 7 in Appendix E bounds the exact KL above and below by explicit multiples of the Fisher quadratic estimate for every finite logit range. With a common finite bound on the logit-error ranges, Corollary 8 extends the same comparison to the mean, every quantile, and CVaR. The guarantees therefore cover both typical errors and the upper tail; how closely the quadratic tracks the exact KL depends on the size of the logit range.

2.3 Propagation-Aware Diagnostics

Our perturbation–propagation theory motivates diagnostics at two levels: an inexpensive layer proxy for selecting layers to restore in causal tests, and a measured module impact for precision allocation. On the FP trajectory $\{x_\ell\}$, a single forward pass reads off each block's local perturbation $\xi_\ell(x_\ell) = \hat{B}_\ell(x_\ell) - x_{\ell+1}$, and a deliberately coarser, quantization-free gain proxy $\widetilde{G}_\ell$ measures $\|\Delta z\|/\|\text{noise}\|$ for unit-norm noise injected at the input of block ℓ, averaged over random directions. The product of the local perturbation magnitude and gain proxy, the butterfly score $b_\ell \coloneqq \widetilde{G}_\ell \cdot \mathbb{E}_x[\|\xi_\ell(x)\|]$, is an inexpensive layer-level proxy for the single-source impact. The butterfly score differs from the local-only score $\mathbb{E}_x[\|\xi_\ell(x)\|]$ only by the gain weighting. The gain weighting isolates the value of propagation, and the resulting score ranks layers for the restoration experiment in Sec. 3.2.

At module level, PAPA uses the measured single-source impact $I_{\ell,m} \coloneqq \mathbb{E}_{x \in \mathrm{hard}}[D_{\mathrm{KL}}]$, obtained by quantizing only module m of layer ℓ. Forward-only, one-component output KL has recently been used to rank average sensitivity for mixed-precision hybrids of state-space models and Transformers (Kong et al., 2026); our butterfly-impact conditions that measurement on a predeclared hard subset and uses the conditioned measurement as the operational source-to-tail proxy in Sec. 4. Tail conditioning plus the derivation and validation of the conditioned score inside the directional, decision-linked framework turn the single-module KL measurement into a reliability-allocation criterion. Our matched-input audit shows that local error alone does not determine propagated failure. Across 256 held-out WikiText-2 windows from Qwen2.5-3B under round-to-nearest (RTN) 4-bit weight quantization, mean per-block local energy has essentially no rank association with mean output KL, at Spearman $\rho = 0.012$. Appendix H.4 records the proxy caveats, the auxiliary module-level scores, the input-level quantization butterfly index (QBI), and the matched-input audit behind the reported correlation (Figure 4).

3 Experiments

Experimental setup. We evaluate pretrained Transformers from under 1B to 72B parameters on a design grid spanning root-mean-square normalization (RMSNorm) or layer normalization

(LayerNorm); grouped-query attention (GQA) or multi-head attention (MHA); and the Swish-gated linear unit (SwiGLU), Gaussian error linear unit (GELU), or rectified linear unit (ReLU). We use Wb to denote b-bit weight-only quantization throughout. Experimental details, including models, quantizers, data, metrics, and evaluation protocols, are provided in Appendix H.

3.1 The Quantization Butterfly Phenomenon

We measure two linked consequences of input-dependent error amplification: the upper tail of output divergence and the early branching of autoregressive generation. Table 4 in Appendix D reports representative cells; the full matrix and the tail-ratio comparison between GPTQ and RTN follow there. We observe a pronounced upper tail even under GPTQ-W4. On Qwen2.5-3B, the mean PPL gap is only 7.3%, yet the p99 per-token KL is 8.9$\times$ the mean. The p99-to-mean ratio's bootstrap 95% confidence interval (CI) is $[8.7, 9.2]\times$ over 1.3×10^5 positions, and the top-1 prediction changes on 14.8% of positions. The effect persists across 7 independent GPTQ calibration draws on Qwen2.5-3B and 3 on Mistral-7B (Appendix M). The small mean PPL gap therefore hides how often the FP and quantized models choose different next tokens.

An independent per-token evaluation in Figure 2(a) shows the same upper-tail concentration. Appendix D resolves that survival curve into one real GPTQ-W4 context in which the same phrase and byte-pair encoding (BPE) continuation occur once as a large KL spike and once as a benign event (Figure 3). Token identity or rarity alone therefore cannot explain membership in the tail.

Consequences for generated text. We now compare FP and quantized continuations from the same prompt. Each model chooses the most probable next token and uses that token as context for the following step. Pooled over Qwen2.5-1.5B and Qwen2.5-3B, 98% of the 384 continuations per model in the larger null-controlled sweep leave the FP trajectory within 48 tokens under GPTQ-W4. A small weight perturbation can therefore redirect a generated continuation. Once a token changes, later predictions use a different prefix.

Comparison with ordinary numerical noise. Greedy decoding can change even when the model weights stay the same. Changing the batch size, the attention kernel, or the compute precision can flip tokens (He & Lab, 2025). We compare quantization against these ordinary numerical perturbations to isolate the output changes caused by low-bit weights; the full settings and numerical results are in Appendix D.

Early and large divergence. Quantization causes much earlier trajectory splits and much larger output shifts than the numerical controls. Under the numerical controls, the per-model WikiText-2 median first divergence falls between tokens 8 and 42, whereas W4 diverges in 92% or more of trajectories in every model and prompt set, with a median first divergence by token 3 (Table 6 in Appendix D). The difference remains large under teacher forcing, where both models receive the same input tokens and model predictions cannot change the context for later steps. On the same windows, the RTN-W4 flip rate is 4.6 to 13.8 times the flip rate of the benign nulls and the RTN-W4 KL is 16 to 323 times larger (Table 6). Low-bit weights therefore cause large output shifts even when the input tokens are identical. Autoregressive decoding then feeds each changed token into later predictions, carrying the disruption through the continuation. Figure 2(b) plots the corresponding greedy-agreement survival curves, and the null-controlled generation sweep also supplies the W3 generation-divergence entries of Table 5.

3.2 Causal and Architectural Validation

We test directly whether the layers selected by the butterfly score are causally responsible for the tail. For each of Qwen2.5-1.5B, Qwen2.5-3B, Llama-3.2-1B, and Pythia-1.4B, we restore exactly the $k = 4$ selected layers to FP inside an otherwise RTN-W3 model and measure the reduction in upper-tail divergence, with bootstrap CIs over evaluation sequences (Table 1). On the Qwen pair, we observe a large, concentrated effect. Restoring the top butterfly layers reduces the p99 KL gap by up to 83% on the 3B model, whereas on the same two models random or bottom-score layers recover at most 16%, and random-layer recovery on the 3B model is negligible, at 1%. The separation is weaker on Llama-3.2-1B and Pythia-1.4B, where random selection is competitive. A few layers therefore account for much of the tail failure on the Qwen models.

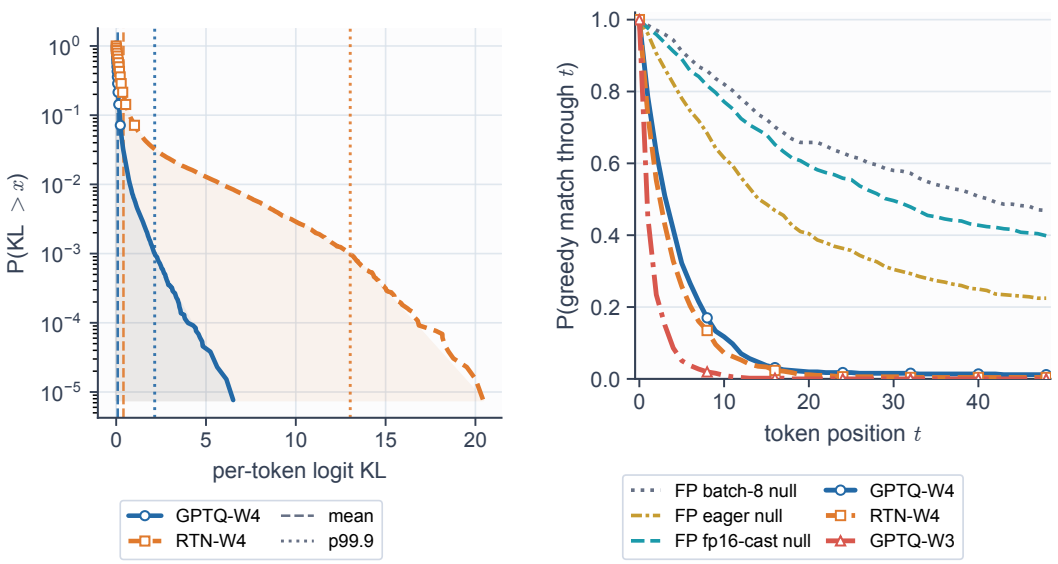


Figure 2: The per-token tail and the trajectory-level consequence. (a) Per-token KL survival on Qwen2.5-3B shows a pronounced upper tail. Survival functions use solid circles for GPTQ-W4 and dashed squares for RTN-W4; vertical lines mark the mean and the 99.9th percentile (p99.9). The independent visualization run uses 256 512-token windows and 64 GPTQ calibration sequences. (b) Greedy agreement with the FP reference, pooled over 512 WikiText-2 prompts from Qwen2.5-1.5B and Qwen2.5-3B. Thin unmarked curves are benign numerical controls, including 16-bit floating-point (fp16) weight casting; heavier marked curves are quantized models and depart from the FP trajectory earlier.

Table 1: Recovery after restoring $k = 4$ layers to FP in an otherwise RTN-W3 model. The per-model columns report p99 KL recovery with sequence-bootstrap 95% CIs. The final two columns report p99 and mean-KL recovery, respectively, averaged across models. The butterfly rows use our layer score, and bold marks the highest recovery in each column.

	recover$_{p99}$ (%)					recover$_{\mu}$ (%)
Layers restored	Qwen2.5-1.5B	Qwen2.5-3B	Llama-3.2-1B	Pythia-1.4B	mean	mean
butterfly top-k	45.6 [43.6, 48.5]	**83.1 [82.7, 83.4]**	**55.0 [53.6, 58.8]**	27.7 [26.7, 31.1]	**52.9**	**54.8**
Hessian top-k	**48.8 [41.3, 50.6]**	67.8 [55.0, 77.5]	19.6 [15.2, 20.9]	14.4 [14.0, 18.0]	37.7	41.0
local-error top-k	43.2 [40.6, 46.5]	76.3 [64.3, 77.6]	15.2 [13.6, 19.4]	14.4 [14.0, 18.0]	37.3	39.7
random-k	16.2 [13.6, 18.1]	0.8 [0.0, 1.8]	41.2 [33.3, 43.4]	**29.6 [23.8, 34.2]**	22.0	22.1
butterfly bottom-k	11.7 [10.2, 17.0]	−0.7 [−0.8, −0.6]	34.3 [31.2, 37.1]	18.7 [17.0, 20.0]	16.0	16.1

Gain-aware selection. Local error measures how much a layer's output changes; downstream gain estimates how much that change affects the model's logits. Including the gain improves which layers we select for restoration. On average across the four models, butterfly selection reduces the p99 KL gap by 53%, compared with 38% for Hessian, 37% for local error, and 22% for random. Where the divergence is concentrated, the CIs of butterfly selection and of the local and second-order selections do not overlap. In our runs, the gain term surfaces early high-gain layers, typically layers 0 to 2, which the local and second-order signals overlook because both signals favor the high-error late layers.

3.3 Tail Failure on Real Tasks

We now test whether the output changes traced by the framework become failures on real requests. Table 19 in Appendix L.1 evaluates six standard zero-shot benchmarks under GPTQ-W4, comparing each quantized model's prediction to the corresponding FP prediction example by example. The benchmarks are HellaSwag, Physical Interaction: Question Answering (PIQA), the Easy and Challenge splits of the AI2 Reasoning Challenge (ARC), WinoGrande, and language modeling broadened

to account for discourse aspects (LAMBADA). We observe little change in average accuracy on most of the suite. Mean accuracy changes by under two points on PIQA and ARC-Easy and even rises slightly on HellaSwag, although LAMBADA and ARC-Challenge each lose about six points. Based on average accuracy alone, the 4-bit quantizer appears nearly lossless on PIQA, ARC-Easy, and HellaSwag.

Per-item failures. However, a large fraction of individual predictions flip, ranging from 7% to 17% across tasks. That interval covers the per-task range; the single worst task–model cell reaches 22% under GPTQ-W4 and 23% under RTN-W4 (Appendix L.1). Across tasks, 4% to 10% of all items suffer harmful flips, in which the FP model gives the correct answer, but the quantized model gives a wrong answer to the same item. WinoGrande and ARC are the most affected: small-margin, reasoning-style items where the decision hinges on a few logits, consistent with the butterfly regime. Beneficial flips mask harmful flips in the average. On HellaSwag, accuracy rises slightly even though 4.2% of items change from correct to wrong. Each prediction flip produces a user-visible change in the multiple-choice answer or the next token. The same model that appears accurate in aggregate changes one prediction in six on WinoGrande. On labeled benchmarks, reporting harmful-flip rates alongside average accuracy makes correct-to-wrong prediction changes visible.

4 Mitigating Quantization-Induced Tail Failures

4.1 PAPA

PAPA uses the decomposition of quantization-induced logit error in Eq. 1 to decide which modules need higher precision. To reduce large output errors under a fixed bit budget, we prioritize modules by the propagated output impact $\|G_{\ell,m}\delta_{\ell,m}\|$, rather than local error alone, $\|\delta_{\ell,m}\|$. To compare precision-allocation scores fairly, we evaluate every method under the same parameter-weighted average bit-width budget, which caps average bits per weight across the quantized modules. Each score ranks modules for higher precision. A shared greedy allocator promotes modules in that order without exceeding the budget. All methods therefore use the same budget limit and differ only in which modules receive higher precision. Local-error scores assign extra bits to modules with large reconstruction errors, whereas butterfly-impact assigns extra bits to modules whose errors cause large output changes (Table 16 in Appendix J).

Measured single-source impact. Because worst-case gain bounds can be loose and rounding is discontinuous (Sec. 2 and Appendix E.1), we measure each module's output impact directly. Specifically, we quantize one module at a time and measure the resulting output KL on a fixed hard subset: the 10% of calibration inputs with the largest KL under uniform quantization. For each module, the mean KL on the hard subset defines the butterfly-impact score. Butterfly-impact extends forward-only component-KL sensitivity (Kong et al., 2026) by focusing the score on inputs most affected by quantization. Computing the butterfly-impact scores requires one calibration sweep, with a cost comparable to collecting GPTQ's Hessians.

Trend across base quantizers. Our cross-base results show that butterfly-impact works best after the base quantizer has reduced local error and made the remaining local errors more similar across modules. Butterfly-impact underperforms local-error baselines on RTN and roughly ties the local-error baselines on the activation-aware weight quantization (AWQ)-style base. On GPTQ, randomized-rotation GPTQ, and learned-rotation GPTQ, butterfly-impact improves hard-subset KL by 8%, 17%, and 18%, respectively, relative to the best competing score (Table 16 in Appendix J). The allocation rule, the full comparison of precision-allocation scores, calibration replication, cost–reliability frontier, family dependence, and 7B results are in Appendix J.

4.2 Reducing Quantization-Induced Output Errors with BART

PAPA assigns extra bits to selected modules to reduce local quantization errors in the selected modules. BART uses the same quantization-error decomposition to reduce output error at a fixed bit-width. BART keeps the deployment precision unchanged and learns a weight correction based on the error that reaches the final output.

Method. For every target linear module, BART freezes the FP weight W and adds a rank-16 low-rank adaptation (LoRA) term ΔW (Hu et al., 2022) plus a per-output-channel learnable clip ratio on the group absmax. The deployed artifact is the single group-wise quantized tensor $\mathrm{quant}(W + \Delta W)$ at the baseline bit-width, matching the PTQ baseline's storage size and inference kernel. Fewer than 2% of the parameters are trained, for 500 steps on 256 calibration sequences.

Objective. BART minimizes the knowledge-distillation (KD) loss, the forward KL between the FP and quantized next-token distributions, $\mathcal{L}_{\mathrm{KD}} = \mathbb{E}_t[\mathrm{KL}(p_t^{\mathrm{FP}} \| p_t^{\hat{\theta}})]$, which locally approximates a Fisher-weighted quadratic in the propagated logit error. The KD loss therefore penalizes downstream consequences rather than only local reconstruction error. An optional CVaR term on per-token KL (Rockafellar et al., 2000) weighted by λ moves the objective from the mean of that divergence toward the divergence tail, tracing a mean-to-upper-tail Pareto curve; the $\lambda = 0$ core carries most of the gain. Theorem 9 explains when a low-rank correction can remove most of the propagated quantization error. With the Jacobian and Fisher matrix fixed, a rank-r correction removes the r largest Fisher-weighted error modes after the module's quantization error propagates to the logits. If the leading Fisher-weighted modes carry most of the error energy, the correction removes most of the quadratic output loss. A local reconstruction with the same rank selects error modes before propagation and can leave the most amplified errors uncorrected, making local reconstruction arbitrarily suboptimal (Proposition 11).

Recovering reasoning at W3. BART recovers reasoning accuracy that GPTQ-W3 nearly eliminates on Qwen2.5-3B-Instruct. GPTQ-W3 reduces Grade School Math 8K (GSM8K) CoT accuracy from 0.847 at FP to 0.030. BART raises GSM8K CoT accuracy to 0.350 while cutting the harmful-flip rate from 82% to 51% (Table 11 in Appendix G). The correction therefore recovers correct multi-step answers while the deployed model stays at 3 bits. Full-capacity quantization-aware training (QAT) with EfficientQAT (Chen et al., 2025) reaches 0.443 on the same protocol while training 90× more parameters. Adding the CVaR penalty targeting the largest 5% of per-token KL values to BART yields $\mathrm{BART}_{\mathrm{tail}}$, which captures 82% of EfficientQAT's accuracy gain over GPTQ-W3. Appendix G gives the low-rank theory, the matched-capacity local baseline, the full-QAT comparison, ablations, and scale results.

5 Discussion and Conclusion

Our work establishes tail reliability as a distinct design target for low-bit LLM outputs. The goal is to keep output errors small and token decisions stable on the inputs most affected by quantization. Input-dependent amplification of quantization error drives the quantization butterfly effect. Small local errors grow into large output shifts on particular requests, creating failures hidden by average metrics, and token changes redirect subsequent generation. Our principal contribution is a complete source-to-decision chain. We follow each realized module- and input-specific error from the source module's output through the directional gain that amplifies or suppresses the error. We bound the resulting output-distribution change, connect the error exactly to a source-space decision radius and a sharp multiclass KL barrier, and extend the analysis to greedy-trajectory stability. This source-to-decision chain goes beyond prior studies that audit answer churn or track average accumulated error. The source-to-decision framework lets us trace rare discrete failures to the responsible modules and inputs and test the links between modules and failures by intervening on the implicated modules.

Empirically, under low-bit LLM quantization the source-to-decision chain exposes a pronounced upper tail of KL divergence, token flips, and autoregressive bifurcations across model families, quantizers, tasks, and KV caches (Appendix K), despite benign average metrics and after matched numerical-null controls. Restoring only the four layers with the highest butterfly scores to FP inside an otherwise 3-bit Qwen2.5-3B model reduces the p99 divergence gap by up to 83%, whereas restoring four random layers reduces that gap by only 1%. The tail failures originate in the layers that the butterfly score predicts. Building on the same propagation analysis, PAPA and BART provide complementary interventions at a fixed bit budget and a fixed bit-width, respectively. Appendix O.1 gives practical guidance for selecting and validating PAPA and BART.

AI USE STATEMENT

Generative AI tools were used as assistive tools for writing and editing, mathematical proofreading and consistency checking, and code development and debugging. Any AI-assisted outputs incorporated into this work were reviewed and validated by the authors as appropriate. The authors take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work evaluates quantized language models using existing models and public benchmarks. The study involves no human subjects, new data collection, or personally identifiable information. We do not anticipate significant ethical concerns specific to this study.

REPRODUCIBILITY STATEMENT

The experimental protocols, including the models, quantizers, data, metrics, calibration procedures, evaluation settings, allocation budgets, and statistical reporting, are specified in Sec. 3 and Appendix H. The BART parameterization, objective, optimizer, training schedule, and other hyperparameters are detailed in Appendix G.5. Replication settings and calibration-draw robustness analyses are reported in Appendix M.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. In *EMNLP*, pp. 4895–4901, 2023.

Yamato Arai and Yuma Ichikawa. Quantization Error Propagation: Revisiting Layer-Wise Post-Training Quantization. In *NeurIPS*, 2025.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-Free 4-Bit Inference in Rotated LLMs. In *NeurIPS*, 2024.

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program Synthesis with Large Language Models. *CoRR*, abs/2108.07732, 2021.

Sirine Ayadi, Sándor Daróczi, Stephan Günnemann, and Bertrand Charpentier. Reliability Scaling Laws for Quantized Large Language Models. *Trans. Mach. Learn. Res.*, 2026.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. In *ICML*, volume 202, pp. 2397–2430, 2023.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about Physical Commonsense in Natural Language. In *AAAI*, pp. 7432–7439, 2020.

Ting-Yun Chang, Muru Zhang, Jesse Thomason, and Robin Jia. Why Do Some Inputs Break Low-Bit LLM Quantization? In *EMNLP*, pp. 3410–3429, 2025.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. QuIP: 2-Bit Quantization of Large Language Models With Guarantees. In *NeurIPS*, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian,

Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code. *CoRR*, abs/2107.03374, 2021.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. EfficientQAT: Efficient Quantization-Aware Training for Large Language Models. In *ACL (1)*, pp. 10081–10100, 2025.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *CoRR*, abs/1803.05457, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems. *CoRR*, abs/2110.14168, 2021.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instructiontuned llm. 2023.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. 2006.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. GPT3.int8(): 8-bit Matrix Multiplication for Transformers at Scale. In *NeurIPS*, 2022.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient Finetuning of Quantized LLMs. In *NeurIPS*, 2023.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. SpQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression. In *ICLR*, 2024.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ: Hessian AWare Quantization of Neural Networks With Mixed-Precision. In *ICCV*, pp. 293–302, 2019.

Abhinav Dutta, Sanjeev Krishnan, Nipun Kwatra, and Ramachandran Ramjee. Accuracy is Not All You Need. In *NeurIPS*, 2024.

Mark Embree and Lloyd N Trefethen. Spectra and pseudospectra: The behavior of nonnormal matrices and operators, 2005.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate Quantization for Generative Pre-trained Transformers. In *ICLR*, 2023.

Horace He and Thinking Machines Lab. Defeating nondeterminism in llm inference. *Thinking Machines Lab: Connectionism*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. In *ICLR*, 2021.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the Knowledge in a Neural Network. *CoRR*, abs/1503.02531, 2015.

Junyuan Hong, Jinhao Duan, Chenhui Zhang, Zhangheng Li, Chulin Xie, Kelsey Lieberman, James Diffenderfer, Brian R. Bartoldson, Ajay Kumar Jaiswal, Kaidi Xu, Bhavya Kailkhura, Dan Hendrycks, Dawn Song, Zhangyang Wang, and Bo Li. Decoding Compressed Trust: Scrutinizing the Trustworthiness of Efficient LLMs Under Compression. In *ICML*, volume 235, pp. 18611–18633, 2024.

Sara Hooker, Aaron C. Courville, Yann N. Dauphin, and Andrea Frome. Selective Brain Damage: Measuring the Disparate Impact of Model Pruning. *CoRR*, abs/1911.05248, 2019.

Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. Characterising Bias in Compressed Models. *CoRR*, abs/2010.03058, 2020.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 Million Context Length LLM Inference with KV Cache Quantization. In *NeurIPS*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*, 2022.

Ajay Kumar Jaiswal, Zhe Gan, Xianzhi Du, Bowen Zhang, Zhangyang Wang, and Yinfei Yang. Compressing LLMs: The Truth is Rarely Pure and Never Simple. In *ICLR*, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *CoRR*, abs/2310.06825, 2023.

Jinuk Kim, Marwa El Halabi, Wonpyo Park, Clemens J. S. Schaefer, Deokjae Lee, Yeonhong Park, Jae W. Lee, and Hyun Oh Song. GuidedQuant: Large Language Model Quantization via Exploiting End Loss Guidance. In *ICML*, volume 267, 2025.

Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. SqueezeLLM: Dense-and-Sparse Quantization. In *ICML*, volume 235, pp. 23901–23923, 2024.

Jason Kong, Nilesh Prasad Pandey, Flavio Ponzina, and Tajana Rosing. A KL Lens on Quantization: Fast, Forward-Only Sensitivity for Mixed-Precision SSM-Transformer Models. *CoRR*, abs/2604.13440, 2026.

Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, and Eunhyeok Park. OWQ: Outlier-Aware Weight Quantization for Efficient Fine-Tuning and Inference of Large Language Models. In *AAAI*, pp. 13355–13364, 2024.

Yixiao Li, Yifan Yu, Chen Liang, Nikos Karampatziakis, Pengcheng He, Weizhu Chen, and Tuo Zhao. LoftQ: LoRA-Fine-Tuning-aware Quantization for Large Language Models. In *ICLR*, 2024.

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 *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 On-Device LLM Compression and Acceleration. In *MLSys*, 2024.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. LLM-QAT: Data-Free Quantization Aware Training for Large Language Models. In *ACL (Findings)*, volume ACL 2024, pp. 467–484, 2024a.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM Quantization with Learned Rotations. In *ICLR*, 2025.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen (Henry) Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A Tuning-Free Asymmetric 2bit Quantization for KV Cache. In *ICML*, volume 235, pp. 32332–32344, 2024b.

EN Lorenz. Deterministic nonperiodic flow. *SPIE MILESTONE SERIES MS*, 75:542–542, 1994.

Kelly Marchisio, Saurabh Dash, Hongyu Chen, Dennis Aumiller, Ahmet Üstün, Sara Hooker, and Sebastian Ruder. How Does Quantization Affect Multilingual LLMs? In *EMNLP (Findings)*, volume EMNLP 2024, pp. 15928–15947, 2024.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and Editing Factual Associations in GPT. In *NeurIPS*, 2022.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer Sentinel Mixture Models. In *ICLR (Poster)*, 2017.

AI Meta. Introducing quantized Llama models with increased speed and a reduced memory footprint. *Meta AI Blog*, 2024.

Ryona Noda. Cross-Layer Error Compensation and Finite-Sample Feature-Statistics Matching for Extreme Low-Bit Quantization of Large Language Models. *arXiv preprint arXiv:2607.14630*, 2026.

Deepanshu Pandey, Arnav Chavan, Nahush Lele, Sankalp Dayal, and Deepak Gupta. Jacobian-guided Noise Injection for Quantization Robustness in Large Language Models. *arXiv preprint arXiv:2608.20988*, 2026.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *ACL (1)*, 2016.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. In *AAAI*, pp. 8732–8740, 2020.

Abel Salinas and Fred Morstatter. The Butterfly Effect of Altering Prompts: How Small Changes and Jailbreaks Affect Large Language Model Performance. In *ACL (Findings)*, volume ACL 2024, pp. 4629–4651, 2024.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. OmniQuant: Omnidirectionally Calibrated Quantization for Large Language Models. In *ICLR*, 2024.

Noam Shazeer. GLU Variants Improve Transformer. *CoRR*, abs/2002.05202, 2020.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, 2024.

Llama Team. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783, 2024.

Yujia Tong, Yuxi Wang, Yunyang Wan, Tian Zhang, Junhao Dong, and Jingling Yuan. Does Compression Preserve Uncertainty? A Unified Benchmark for Quantized and Sparse LLMs via Conformal Prediction. *CoRR*, abs/2606.01850, 2026.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey

Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models. *CoRR*, abs/2307.09288, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NIPS*, pp. 5998–6008, 2017.

Zekun Wu, Swati Dhiman, and Adriano Koshiyama. Quantization Damage Is Multiplicative, Not Additive. *arXiv preprint arXiv:2608.06564*, 2026.

Guangxuan Xiao, Ji Lin, Mickaël Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. In *ICML*, volume 202, pp. 38087–38099, 2023.

Bingxin Xu, Zhen Dong, Oussama Elachqar, and Yuzhang Shang. ButterflyQuant: Ultra-low-bit LLM Quantization through Learnable Orthogonal Butterfly Transforms. *CoRR*, abs/2509.09679, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report. *CoRR*, abs/2412.15115, 2024a.

Wanli Yang, Fei Sun, Xinyu Ma, Xun Liu, Dawei Yin, and Xueqi Cheng. The Butterfly Effect of Model Editing: Few Edits Can Trigger Large Language Models Collapse. In *ACL (Findings)*, volume ACL 2024, pp. 5419–5437, 2024b.

Yongyi Yang, Jianyang Gao, and Wei Hu. RaanA: A Fast, Flexible, and Data-Efficient Post-Training Quantization Algorithm. *CoRR*, abs/2504.03717, 2025.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. ZeroQuant: Efficient and Affordable Post-Training Quantization for Large-Scale Transformers. In *NeurIPS*, 2022.

Zhewei Yao, Xiaoxia Wu, Cheng Li, Stephen Youn, and Yuxiong He. Exploring Post-training Quantization in LLMs from Comprehensive Study to Low Rank Compensation. In *AAAI*, pp. 19377–19385, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a Machine Really Finish Your Sentence? In *ACL (1)*, pp. 4791–4800, 2019.

Biao Zhang and Rico Sennrich. Root Mean Square Layer Normalization. In *NeurIPS*, pp. 12360–12371, 2019.

Cheng Zhang, Jianyi Cheng, George Anthony Constantinides, and Yiren Zhao. LQER: Low-Rank Quantization Error Reconstruction for LLMs. In *ICML*, volume 235, pp. 58763–58779, 2024.

Cheng Zhang, Jeffrey T. H. Wong, Can Xiao, George Anthony Constantinides, and Yiren Zhao. QERA: an Analytical Framework for Quantization Error Reconstruction. In *ICLR*, 2025a.

Fred Zhang and Neel Nanda. Towards Best Practices of Activation Patching in Language Models: Metrics and Methods. In *ICLR*, 2024.

Nan Zhang, Yusen Zhang, Prasenjit Mitra, and Rui Zhang. When Reasoning Meets Compression: Benchmarking Compressed Large Reasoning Models on Complex Reasoning Tasks. *CoRR*, abs/2504.02010, 2025b.

Qian Zhang, Yaoming Li, Zhewen Tan, Yanshu Wang, Heng Lu, Kun Su, Zongwei Lv, Wenhan Yu, Yongge Ma, Yinjun Han, et al. REAL-Q: E2E LLM Quantization via Dynamic Gradient Descent. *arXiv preprint arXiv:2609.00049*, 2026.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open Pre-trained Transformer Language Models. *CoRR*, abs/2205.01068, 2022.

Chenxi Zhou, Pengfei Cao, Jiang Li, Bohan Yu, Jinyu Ye, Jun Zhao, and Kang Liu. From Signal Degradation to Computation Collapse: Uncovering the Two Failure Modes of LLM Quantization. In *ACL (Findings)*, pp. 23204–23222, 2026a.

Chenxi Zhou, Pengfei Cao, Jinyu Ye, Bohan Yu, Haida Yu, Jiang Li, Jun Zhao, and Kang Liu. Quantization Degradation in Large Language Models: A Signal-Noise Perspective. *arXiv preprint arXiv:2608.08188*, 2026b.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-Following Evaluation for Large Language Models. *CoRR*, abs/2311.07911, 2023.

Appendix Table of Contents

A List of Notation **15**

B Contribution and Evidence Map **16**

C Expanded Related Work **16**

D Expanded Butterfly-Phenomenon and Numerical-Null Results **20**

E Theoretical Extensions and Formal Statements **21**

F Proofs of the Formal Results **27**

G Training-Time Butterfly Mitigation **31**

H Experimental Details **40**

I Extended Empirical Validation **43**

J Extended Mixed-Precision Allocation Results **47**

K Beyond Weights: KV-Cache Quantization **50**

L Additional Task-Level Analyses **51**

M Replication and Calibration-Draw Robustness **53**

N Input-Adaptive Precision **53**

O Extended Discussion and Limitations **55**

A List of Notation

Table 2: List of notation. Reused symbols are distinguished by the stated context for each symbol.

Symbol	Meaning
Propagation model and diagnostics	
L, T, d, V; $\ell, m, a, \mathcal{A}$	layer count, sequence length, width, vocabulary; source indices and set
$x_\ell, \hat{x}_\ell, u_\ell$	FP and quantized residual states and block intermediate

Symbol	Meaning
$B_\ell, \hat{B}_\ell$	FP and quantized Transformer blocks
$A_\ell, M_\ell, N_\ell, N'_\ell$	attention, MLP, and normalization maps
$\xi_\ell, \delta_{\ell,m}$	block residual and module-source perturbation
$e_\ell, \mathcal{R}_\ell$	state error and recurrence remainder
$J_\ell, J_{\ell,m}$	block and module-to-block Jacobians
$\Phi_{a:b}, C, R$	Jacobian product, readout Jacobian, readout map
$G_{\ell,m} = G_a, v_a$	propagated source map and corresponding logit contribution
$\mathrm{Gain}^{\mathrm{dir}}_{\ell,m}, \mathrm{Impact}_{\ell,m}$	directional source gain and propagated impact
$z, \hat{z}, \Delta z, r$	FP and quantized logits; logit error and exact remainder
$\alpha, \Gamma_\ell, L_{J,\ell}, \kappa_\alpha$	perturbation scale and stability constants
$\bar{E}_\ell, P_{a:b}$	error envelope and worst-case gain product
$\mathrm{osc}(\cdot), \Omega$	logit range and corresponding certificate bound
$p, q, D_{\mathrm{KL}}(p\|q)$	FP and quantized distributions and output divergence
F, $A = \eta^\top F \eta/2$	softmax Fisher matrix and quadratic
$\Pi_\perp, \mathbf{1}$	constant-logit projector; all-ones vector and indicator
$\widetilde{G}_\ell, b_\ell$	gain proxy and butterfly layer score
$H_m, h_m, \Delta W_m, e_{\ell,m}$	calibration moment and input; local error energy
$I_{\ell,m}, \overline{D}_{\mathrm{KL}}, E_{\mathrm{loc}}, \epsilon, \mathrm{QBI}$	impact, mean KL, local energy, input index
Decision and tail guarantees	
c, j, e_j, M_j	winner, competitor, basis vector, and logit margin
G, δ, Λ	concatenated source map and error; source metric
d_j, δ^*_j	directional dual gain and least-norm boundary witness
$G_t, c_t, M_{t,j}, d_{t,j}$	stepwise map, winner, margin, and gain
ρ^*_H, H	minimum trajectory radius and decoding horizon
$a, \kappa(p)$	runner-up and minimum KL flip barrier
$p_t, q_t, D_t, \kappa_t, T$	teacher-forced laws, KL, barrier, first divergence
$\eta, c_-(\Omega), c_+(\Omega)$	logit perturbation and finite-range KL factors
X, A_X, K_X, b	random input, Fisher and KL values, range envelope
$Q_\alpha, \mathrm{CVaR}_\tau, \tau$	quantile, upper-tail risk, tail mass
BART and low-rank correction	
$W, \Delta W$	base weight and BART correction
A, B, r, α (LoRA)	low-rank factors, rank, and scale
h, Σ, E, D	module input, input moment, error, correction
$P = F^{1/2} G$	Fisher-weighted propagated map
$\mathcal{J}, \mathcal{J}_{\mathrm{loc}}$	propagated and local quadratic risks
$M, M_r, U, V, \sigma_j, P^\dagger$	propagated-error singular value decomposition (SVD) quantities and pseudoinverse
$\mathrm{cond}(P)$	propagation condition number
$\mathcal{L}_{\mathrm{KD}}, \mathcal{L}_{\mathrm{CVaR}_q}, \mathcal{L}_{\mathrm{NLL}}$	distillation, tail, and negative log-likelihood (NLL) losses
$\lambda, \beta_{\mathrm{NLL}}, q, \nu$ (BART)	loss weights, tail mass, value-at-risk (VaR) threshold
$S_{i,j}, m_i, a_i, o_i, V_j$	attention score and margin, weights and output, values
Q, γ, S, s_j	rotation, norm scale, Cayley matrix, AWQ scale

B Contribution and Evidence Map

Table 3 maps each main claim to the corresponding direct evidence and location.

C Expanded Related Work

The same butterfly-effect metaphor has been used for large behavioral changes caused by small prompt perturbations or model edits (Salinas & Morstatter, 2024; Yang et al., 2024b); here the perturbation is quantization error, the amplification occurs through Transformer computation and autoregressive feedback, and the observed consequences are the divergence tail and trajectory

Table 3: Compact map from the paper's claims to the supporting evidence.

Claim	Direct evidence	Location
Tail reliability is distinct from mean fidelity	Per-token KL tails, top-1 flips, greedy divergence, task-level harmful flips, and matched FP numerical controls	Secs. 3.1 and 3.3; Appendix D
Directional source-to-decision link	Realized-source decomposition, finite-perturbation tail transfer, exact source-space radius, sharp multiclass KL barrier, and trajectory certificate	Sec. 2; Appendices E and F
Causal localization	Score-guided FP layer restoration against local, curvature, random, and bottom-score controls	Sec. 3.2
PAPA	Matched-budget comparisons across local, curvature, and propagation-aware scores	Sec. 4; Appendix J
Training-time mitigation	BART comparisons at matched bit-widths, capacity controls, ablations, and scale results	Sec. 4.2; Appendix G

bifurcations characterized in Sec. 3.1. The closest quantization literature has established separate pieces of the quantization tail-reliability problem. Audits measure answer churn, vulnerable inputs, decision-margin shrinkage, or uncertainty and robustness under compression (Dutta et al., 2024; Chang et al., 2025; Wu et al., 2026; Tong et al., 2026; Ayadi et al., 2026). Propagation and optimization work tracks accumulated error, Jacobian or output sensitivity, end-loss guidance, single-component KL, or global average fidelity (Arai & Ichikawa, 2025; Yang et al., 2025; Kim et al., 2025; Kong et al., 2026; Zhou et al., 2026b;a; Noda, 2026; Zhang et al., 2026; Pandey et al., 2026; Zhang et al., 2025b). However, the cited audits and quantization methods do not jointly connect a realized quantization source to the upper tail, an exact discrete decision boundary, the ensuing autoregressive trajectory, and a causal intervention. To our knowledge, our framework is the first unified source-to-decision account of that chain. The framework's three inseparable elements are tail reliability measured against matched numerical nulls, input- and direction-specific propagation $G_{\ell,m}\delta_{\ell,m}$, and exact links from the propagated perturbation to KL, top-1 decisions, and greedy trajectories, closed by causal localization and tail-targeted mitigation.

Modern PTQ largely minimizes local or blockwise reconstruction error, often weighted by curvature or activation sensitivity, while rotation methods reduce outlier alignment (Frantar et al., 2023; Lin et al., 2024; Shao et al., 2024; Ashkboos et al., 2024; Liu et al., 2025). A growing output-aware line instead injects end-loss guidance, ranks single-component output KL, compensates accumulated error, or directly optimizes network-level fidelity (Kim et al., 2025; Kong et al., 2026; Arai & Ichikawa, 2025; Noda, 2026; Zhang et al., 2026). Outlier-aware mixed precision keeps sensitive weights or channels in higher precision (Lee et al., 2024; Dettmers et al., 2024; Kim et al., 2024), while QAT, distillation, and low-rank compensation retrain weights or learn compact corrections (Liu et al., 2024a; Chen et al., 2025; Yao et al., 2024; Zhang et al., 2024; Dettmers et al., 2023). The surveyed

quantization methods establish increasingly global ways to preserve average fidelity. Our object is different: the upper tail of FP–quantized divergence generated by each realized, directional source $G_{\ell,m}\delta_{\ell,m}$ and the exact discrete consequences of the propagated source. Thus the methods above are both bases and strong controls for our source-to-decision analysis.

Prior compression audits expose example-level flips despite similar aggregate quality (Hooker et al., 2019; Jaiswal et al., 2024; Dutta et al., 2024). Closest to our empirical starting point, Dutta et al. (2024) use flips and KL to document divergence at matched accuracy, while Chang et al. (2025) identify inputs repeatedly harmed across low-bit methods and associate those inputs with residual magnitudes, late layers, and MLP gates. Wu et al. (2026) fit bit-dependent shrinkage of two-choice margins and accurately predict aggregate and per-decision flip probabilities. Broader reliability studies evaluate uncertainty, calibration, and robustness to input perturbations under quantization (Tong et al., 2026; Ayadi et al., 2026). Uncertainty, calibration, and robustness are important but different reliability coordinates. Our target is the controlled upper tail of the FP–quantized output-distribution discrepancy, across whole-vocabulary tokens and free-running trajectories. More importantly, we connect that tail to the realized internal source direction, an exact source-space radius in the propagated model, a sharp multiclass KL decision barrier, causal restoration, and tail-targeted mitigation. In particular, the margin study is empirical and first-decision focused and does not establish a causal path from a perturbed weight to the moved margin; our theory and interventions supply that missing source-to-decision path.

The closest mechanistic and intervention baselines fall into three groups. Propagation-aware PTQ compensates accumulated cross-layer error (Arai & Ichikawa, 2025); RaanA uses an output-Jacobian norm for layer-wise bit allocation (Yang et al., 2025); GuidedQuant imports end-loss gradients into a local objective (Kim et al., 2025); and a forward-only KL lens ranks one-at-a-time component perturbations for mixed precision (Kong et al., 2026). Concurrent work decomposes degradation into source noise and scalar cross-layer propagation, optimizes network-level accumulated features or KL, or regularizes local attention-Jacobian sensitivity (Zhou et al., 2026b; Noda, 2026; Zhang et al., 2026; Pandey et al., 2026). Other compression-specific analyses separate smooth signal degradation from 2-bit computation collapse or use attribution patching to protect reasoning-critical weights (Zhou et al., 2026a; Zhang et al., 2025b). These approaches advance accumulation, allocation, causal localization, or average fidelity. We preserve the realized input- and module-specific direction through the end-to-end logit map $G_{\ell,m}\delta_{\ell,m}$, condition the empirical proxy for the propagated impact on the hard subset, and ask whether that direction reaches a rare output and decision tail. The exact decision radius and KL barrier, distribution-level tail guarantees, and causal restoration are the additional links that turn propagation sensitivity into a source-to-decision reliability framework. Hessian-based allocation assigns mixed precision by local curvature (Dong et al., 2019), and KV cache quantizers adopt per-channel keys and per-token values (Hooper et al., 2024; Liu et al., 2024b), a design that our routing–content analysis explains. On the analysis side, transient growth in non-normal operators motivates studying directional gain rather than local magnitude (Embree & Trefethen, 2005). On the intervention side, we repurpose activation patching to test whether score-flagged layers are causally responsible for tail failures (Meng et al., 2022; Zhang & Nanda, 2024). Finally, the butterfly-effect metaphor already appears in LLM studies of prompt perturbations and model editing (Salinas & Morstatter, 2024; Yang et al., 2024b); we use the metaphor to name the quantization-specific empirical signature. Our contribution is the tail-reliability, source-to-decision characterization and causal mitigation of that signature. ButterflyQuant (Xu et al., 2025) instead refers to a structured orthogonal transform and is unrelated to the phenomenon studied here. The remainder of this section gives the full comparison.

PTQ. PTQ commonly optimizes local or blockwise reconstruction. Li et al. (2021) introduced block reconstruction, Yao et al. (2022) used layer-wise distillation, and Frantar et al. (2023) introduced GPTQ with approximate second-order layer-wise reconstruction. AWQ and SmoothQuant (Lin et al., 2024; Xiao et al., 2023) protect or migrate salient activation channels; OmniQuant (Shao et al., 2024) learns clipping and equivalent transforms; and QuIP, QuaRot, and SpinQuant (Chee et al., 2023; Ashkboos et al., 2024; Liu et al., 2025) use rotations to reduce alignment with sensitive directions. Outlier-aware mixed precision retains locally sensitive weights or channels (Lee et al., 2024; Dettmers et al., 2024; Kim et al., 2024). GuidedQuant extends local reconstruction with end-loss gradient structure (Kim et al., 2025), and REAL-Q directly optimizes end-to-end KL (Zhang et al., 2026). The PTQ methods optimize local, guided, or global fidelity and typically select by mean objectives;

we study the directional source $G_{\ell,m}\delta_{\ell,m}$ and the distributional tail that source produces. The PTQ methods are therefore natural baselines and plug-in targets for our analysis.

Training-time QAT and low-rank compensation. QAT and data-free quantization-aware training for large language models (LLM-QAT) (Liu et al., 2024a) retrain weights to absorb quantization error, usually through reconstruction or distillation (Hinton et al., 2015). Low-rank compensation is likewise established: ZeroQuant-V2, LQER, and QERA construct low-rank corrections (Yao et al., 2024; Zhang et al., 2024; 2025a), while QLoRA and LoftQ combine quantized backbones with low-rank adapters (Dettmers et al., 2023; Li et al., 2024). BART uses the lightweight LoRA parameterization of Hu et al. (2022) with learnable clips, trained by distillation plus a Rockafellar–Uryasev CVaR term (Rockafellar et al., 2000). BART's contribution is to align a fixed-bit low-rank correction with the propagated error geometry and optimize the same tail-reliability target exposed by our source-to-decision analysis, rather than local reconstruction error.

Example-level reliability under compression. Prior work shows that the effects of compression are not uniform across examples: Pruned or quantized vision models can sacrifice a small set of exemplars while preserving top-line accuracy (Hooker et al., 2019; 2020), and compressed LLMs can lose task quality that PPL hides (Jaiswal et al., 2024). Compression audits also identify effects on trustworthiness and multilingual behavior (Hong et al., 2024; Marchisio et al., 2024). Most closely, Dutta et al. (2024) use flips and KL to expose large divergence between compressed and baseline models despite matched accuracy, while Chang et al. (2025) show that low-bit methods disproportionately harm the same inputs and connect the resulting errors to residual-stream magnitudes, late layers, and MLP gates. Wu et al. (2026) measure multiplicative shrinkage of binary-choice margins across bit-widths and predict which forced-choice decisions flip. Quantization reliability has also been studied through conformal uncertainty, calibration, and robustness to semantics-preserving input perturbations (Tong et al., 2026; Ayadi et al., 2026). The compression audits and reliability studies establish answer churn, input heterogeneity, margin contraction, and distinct uncertainty axes. We isolate the upper tail of the FP–quantized whole-vocabulary distribution shift under matched numerical controls, extend that tail through free-running trajectories, and connect the tail to directional source propagation, exact decision links, causal interventions, and two tail-targeted mitigations.

Error propagation and precision allocation. The method of Arai & Ichikawa (2025) is the closest propagation-based PTQ work and compensates cross-layer accumulated error during layer-wise quantization. Yang et al. (2025) use output-aware layer sensitivity for bit allocation, Kim et al. (2025) use end-loss guidance, and Kong et al. (2026) use one-component output KL for forward-only mixed-precision ranking. Concurrent analyses decompose quantization degradation into source noise and scalar cross-layer propagation, optimize network-level accumulated features or KL, or regularize local attention Jacobian sensitivity (Zhou et al., 2026b; Noda, 2026; Zhang et al., 2026; Pandey et al., 2026). Complementary work distinguishes signal degradation from 2-bit computation collapse and applies mechanism-specific repair (Zhou et al., 2026a). The propagation and repair methods target accumulation, allocation, local sensitivity, collapse regimes, or average end-to-end fidelity. We instead retain module-wise, input-dependent realized directions $G_{\ell,m}\delta_{\ell,m}$, characterize which sources reach the output tail, and carry those sources to exact decision and trajectory conditions before testing the same sources with single-source interventions and causal restoration. Hessian-aware quantization (HAWQ) (Dong et al., 2019) is a strong mixed-precision baseline based on local Hessian sensitivity; unlike our score, HAWQ does not model downstream gain or tail risk. KVQuant and KIVI (Hooper et al., 2024; Liu et al., 2024b) empirically adopt per-channel keys and per-token values for KV-cache compression; our routing–content analysis explains this asymmetric design rule.

Non-normal amplification and causal verification. Prior analyses of finite-depth transient growth in non-normal operators motivate the study of directional downstream gain rather than local perturbation magnitude alone (Embree & Trefethen, 2005). Causal and activation patching provide the intervention tools that we repurpose to test whether score-flagged modules are causally responsible for quantization tail failures (Meng et al., 2022; Zhang & Nanda, 2024). Compression-specific attribution patching has identified final-layer and MLP gate weights important to average reasoning accuracy and protected a small weight subset (Zhang et al., 2025b). Our intervention target is instead the distributional tail predicted by directional propagation, and the restoration experiment tests whether score-selected paths causally carry that tail.

Table 4: Representative quantization-butterfly cells. A small mean PPL gap can coexist with a pronounced upper tail of per-token KL, frequent top-1 flips, and divergent greedy trajectories. ΔPPL is the mean PPL gap over FP; KL_μ and KL_{p99} are the mean and p99 of the per-token logit $D_{\mathrm{KL}}(p_{\mathrm{FP}} \| p_{\mathrm{Q}})$; the tail column is the ratio $\mathrm{KL}_{p99}/\mathrm{KL}_\mu$; the flip column is the teacher-forced per-token top-1 disagreement rate; and the gen column is the fraction of greedy generations that diverge from the FP trajectory within 48 tokens, here on a 24- to 32-prompt diagnostic. The full model–quantizer matrix and numerical-null controls are in Tables 5 and 6.

Model	setting	ΔPPL	KL_μ	KL_{p99}	tail	flip	gen
Llama-3.1-8B	GPTQ-W4	5.7%	0.065	0.69	10.6×	11.4%	100%
Mistral-7B	GPTQ-W4	3.9%	0.040	0.36	9.0×	8.8%	94%
Qwen2.5-3B	GPTQ-W4	7.3%	0.102	0.91	8.9×	14.8%	100%
Qwen2.5-3B	RTN-W4	42.7%	0.417	5.81	13.9×	26.0%	100%

Terminology. The butterfly effect is an established metaphor for sensitive dependence, and prior LLM work uses the metaphor for prompt perturbations and model editing (Salinas & Morstatter, 2024; Yang et al., 2024b). Here the term connects small rounding errors to amplified output consequences: large divergence on particular inputs and early branching of autoregressive trajectories. Average metrics hide this input-dependent amplification and the harmful decisions produced by the amplification. The technical advance is the unified tail-reliability, directional-propagation, and exact-decision framework that explains and mitigates these consequences. ButterflyQuant (Xu et al., 2025) is unrelated terminology: The ButterflyQuant name refers to the structured orthogonal transform used by the quantizer.

D Expanded Butterfly-Phenomenon and Numerical-Null Results

This appendix expands Sec. 3.1. Table 4 gives representative cells, and Table 5 expands the representative cells to the full model–quantizer matrix, spanning three families from 1.5B to 70B at 3 and 4 bits; every cell shows the same signature of a small mean gap beside a pronounced upper tail of per-token KL. Table 6 then reports the per-condition numbers behind Figure 2(b), in which the benign FP-to-FP perturbations decay slowly and stay near the top whereas every quantized setting departs within a few tokens. Figure 3 finally resolves the per-token survival curve of Figure 2(a) into one real context.

Generation diagnostics and numerical controls. Both generation diagnostics below evaluate Qwen2.5-1.5B and Qwen2.5-3B. The smaller 24- to 32-prompt diagnostic gives 97% and 100% on the same two models (Table 5).

We therefore re-measure generation divergence at scale using 384 real prompts per model: 256 WikiText-2 prefixes and 128 Dolly-15k instructions. We decode 48 greedy tokens against the same FP reference, run with batch 1, the scaled dot-product attention (SDPA) kernel, and bf16. We compare three benign FP-to-FP perturbations, namely batch-8 execution, the eager-attention kernel, and fp16 weight casting, against four quantized settings (Table 6). Even these numerical controls change generated text. Batch-8 execution alone eventually changes 53% of the WikiText-2 trajectories. A final mismatch alone does not reveal when the paths split or how large the output change was.

Tail ratio under a stronger quantizer. In five of the six model×bit-width pairs for which both quantizers were measured (Table 5), the tail ratio is higher for GPTQ than for RTN, because GPTQ shrinks the mean more than the tail. Improving the average therefore does not eliminate the tail. Qwen2.5-3B at W4 is the single exception, because RTN has much larger overall divergence there, which raises the RTN tail ratio as well.

Inside a real tail event. Figure 3 resolves the survival curve of Figure 2(a) into one real GPTQ-W4 context. Among 256 held-out windows, the displayed window is selected deterministically as the window whose five largest positions carry the largest fraction of the window's total KL. Within that window the same phrase and BPE continuation occur twice, once as a large KL spike and once as

Table 5: Quantization-butterfly results across models and bit-widths. The tail column is KL p99 divided by mean KL, the flip column is teacher-forced top-1 disagreement, and the gen column is greedy divergence within 48 tokens. Rates are percentages. The W4 gen column uses the 24- to 32-prompt diagnostic and the W3 gen column the 384-prompt null-control sweep; a dash marks a cell that was not measured.

Model	setting	ΔPPL (%)	KL_μ	KL_{p99}	tail ($\times$)	flip (%)	gen (%)
Llama-3.1-8B	GPTQ-W3	31.7	0.319	2.96	9.3	24.2	–
	RTN-W3	72.5	0.575	3.77	6.6	34.5	–
	GPTQ-W4	5.7	0.065	0.69	10.6	11.4	100
	RTN-W4	8.7	0.092	0.80	8.7	14.2	97
Llama-3.3-70B-Instruct	RTN-W4	78.7	0.480	7.03	14.7	23.2	100
Mistral-7B	GPTQ-W3	68.0	0.411	3.62	8.8	25.7	–
	GPTQ-W4	3.9	0.040	0.36	9.0	8.8	94
Qwen2.5-1.5B	GPTQ-W3	95.5	0.798	5.08	6.4	39.2	100
	RTN-W3	674.7	2.120	7.90	3.7	63.4	100
	GPTQ-W4	8.3	0.118	1.02	8.6	16.3	97
	RTN-W4	17.1	0.205	1.46	7.1	22.1	100
Qwen2.5-3B	GPTQ-W3	77.5	0.686	4.91	7.2	35.1	100
	RTN-W3	4.41×10^5	8.323	23.19	2.8	97.1	100
	GPTQ-W4	7.3	0.102	0.91	8.9	14.8	100
	RTN-W4	42.7	0.417	5.81	13.9	26.0	100
Qwen2.5-7B	RTN-W4	8.3	0.116	0.81	7.0	16.1	100
Qwen2.5-14B	RTN-W3	118.1	0.917	5.38	5.9	40.3	–
	RTN-W4	16.6	0.207	2.43	11.7	17.7	100
Qwen2.5-32B	RTN-W4	10.6	0.132	1.41	10.7	14.5	100
Qwen2.5-72B	RTN-W4	10.8	0.117	1.48	12.7	12.0	100

Table 6: Quantization relative to benign numerical controls, averaged over the two Qwen2.5 models. Greedy statistics use 256 WikiText-2 and 128 Dolly-15k prompts per model; teacher-forced statistics use 64 windows of 512 tokens. All conditions are scored against the same FP reference, run at batch 1 with the SDPA kernel in bf16. The teacher-forced columns were measured for the three nulls and for RTN-W4; a dash marks a cell that was not measured.

	gen-div (%)		median 1st-div	teacher-forced	
Condition	WikiText-2	Dolly-15k	(WikiText-2)	flip (%)	KL
FP, batch 8 (null)	53	32	40	1.74	1.04×10^{-3}
FP, eager kernel (null)	78	44	19	5.23	1.95×10^{-2}
FP, fp16 cast (null)	60	37	30	2.26	9.90×10^{-4}
GPTQ-W4	99	95	2	–	–
RTN-W4	100	93	2	23.96	3.20×10^{-1}
GPTQ-W3	100	100	0	–	–
RTN-W3	100	100	0	–	–

a benign event. The selected window does not estimate a typical effect size; the within-window comparison shows that token identity or rarity alone cannot explain membership in the tail.

E Theoretical Extensions and Formal Statements

E.1 Propagation Model and First-Order Remainder Bound

Setup. A Transformer block (Vaswani et al., 2017) maps the residual stream $x_\ell \in \mathbb{R}^{T\times d}$ through

$$u_\ell \coloneqq x_\ell + A_\ell(N_\ell(x_\ell)), \qquad x_{\ell+1} \coloneqq u_\ell + M_\ell(N'_\ell(u_\ell)); \quad B_\ell(x_\ell) \coloneqq x_{\ell+1}.$$

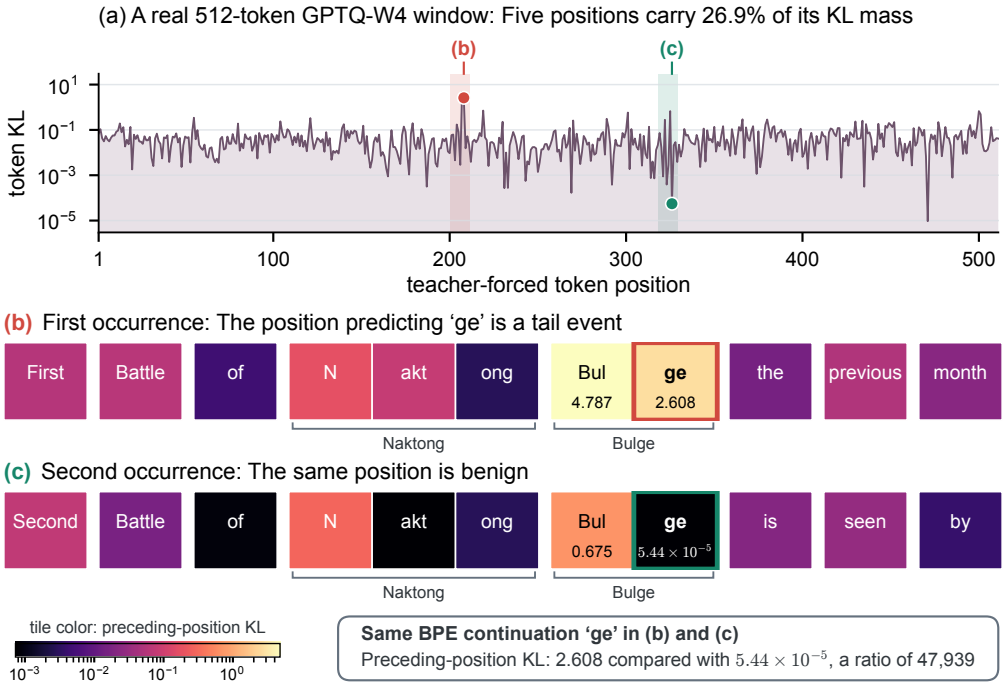


Figure 3: The same continuation can be benign or land in the KL tail. The shown Qwen2.5-3B GPTQ-W4 WikiText-2 window maximizes, among 256 windows, the fraction of total KL carried by the window's five largest positions; the five highest-KL positions carry 26.9% of the window's KL mass. The phrase Naktong Bulge occurs twice. For the same final BPE continuation `ge`, the preceding position's $D_{\mathrm{KL}}(p_{\mathrm{FP}} \| p_{\mathrm{Q}})$ is 2.608 in the first occurrence and 5.44×10^{-5} in the second occurrence. Lexical identity is therefore insufficient: The surrounding context must enter the explanation of a tail event.

Here N_ℓ, N'_ℓ denote RMSNorm (Zhang & Sennrich, 2019) or LayerNorm, A_ℓ denotes attention, and M_ℓ denotes the MLP. A quantized block differs from the corresponding FP block by the input-dependent residual

$$\xi_\ell(x) := \hat{B}_\ell(x) - B_\ell(x), \qquad \hat{B}_\ell(x) = B_\ell(x) + \xi_\ell(x),$$

which collects weight, activation, and KV quantization error.

Error recurrence. With $e_\ell := \hat{x}_\ell - x_\ell$ and the quantization strength treated as a small parameter, expansion around the FP trajectory gives

$$e_{\ell+1} = J_\ell(x_\ell) e_\ell + \xi_\ell(x_\ell) + \mathcal{R}_\ell, \qquad J_\ell(x_\ell) := DB_\ell(x_\ell).$$

Here $\mathcal{R}_\ell$ contains both block curvature and the input drift of the quantization residual. Under the conditions of Lemma 3 below, the recurrence remainder obeys $\|\mathcal{R}_\ell\| \le (L_{J,\ell}/2)\|e_\ell\|^2 + \kappa_\alpha \|e_\ell\|$. Because real rounding quantizers are discontinuous at thresholds, the expansion serves as a local surrogate. Writing the FP logits as $z := R(x_L)$, define $\Phi_{a:b} := J_{b-1} \cdots J_a$ and the readout Jacobian $C := DR(x_L)$. The two trajectories start from the same residual state, so $e_0 = 0$. Define $\Delta z := \hat{z} - z$ as the change from FP logits z to quantized logits $\hat{z}$. Unrolling the recurrence approximates Δz by summing each block's error after propagation through the remaining network to the output:

$$\Delta z \approx C \sum_{\ell=0}^{L-1} \Phi_{\ell+1:L} \xi_\ell(x_\ell).$$

To identify which modules contribute to Δz, we decompose each block residual into module sources

$$\xi_\ell \approx \sum_m J_{\ell,\mathrm{after}(m)} \delta_{\ell,m},$$

where $J_{\ell,\mathrm{after}(m)}$ maps a perturbation at module m's output to the block output. Write $J_{\ell,m} \coloneqq J_{\ell,\mathrm{after}(m)}$ for $m \in \{\mathrm{Q,K,V,O,gate,up,down}\}$. Substituting the module-source decomposition into the logit expansion gives the module-wise perturbation–propagation decomposition in Eq. 1.

The following standard-form bound specifies the regime in which the first-order term is trustworthy; Appendix F.1 proves the remainder bound.

Lemma 3 (First-order remainder bound)**.** *Consider a family of perturbations indexed by $\alpha \to 0^+$. Suppose, on the line segments joining the FP and perturbed states, that each block Jacobian satisfies $\|J_k\|_{\mathrm{op}} \le \Gamma_k$ and is $L_{J,k}$-Lipschitz, while $\|\xi_k(x)\| \le \alpha$ and $\|D\xi_k(x)\|_{\mathrm{op}} \le \kappa_\alpha$. Let $P_{a:b} \coloneqq \prod_{k=a}^{b-1} \Gamma_k$, where an empty product equals one, and define the nonlinear error envelope recursively by*

$$\bar{E}_0 \coloneqq 0, \qquad \bar{E}_{k+1} \coloneqq \alpha + (\Gamma_k + \kappa_\alpha)\bar{E}_k + \tfrac{L_{J,k}}{2}\bar{E}_k^2.$$

Then $\|e_k\| \le \bar{E}_k$ and

$$\left\| e_L - \textstyle\sum_{\ell=0}^{L-1} \Phi_{\ell+1:L}\xi_\ell(x_\ell) \right\| \le \displaystyle\sum_{\ell=0}^{L-1} P_{\ell+1:L}\left[\tfrac{L_{J,\ell}}{2}\bar{E}_\ell^2 + \kappa_\alpha \bar{E}_\ell\right].$$

The bound separates two ways the first-order approximation can lose accuracy: Block curvature contributes terms quadratic in the state error, while changes in the quantization residual contribute terms proportional to $\kappa_\alpha\|e_\ell\|$. Both are amplified by the remaining blocks, so small injected errors alone do not ensure an accurate approximation. For fixed depth and uniformly bounded constants, the recursion gives $\bar{E}_k = O(\alpha)$. Hence $\kappa_\alpha = o(1)$ makes the resulting remainder $o(\alpha)$, yielding an $o(\alpha)$-accurate first-order expansion. If only $\kappa_\alpha = O(1)$ is known, the bound permits an $O(\alpha)$ input-drift remainder, and the bound alone does not give $o(\alpha)$ accuracy.

Worst-case products $P_{a:b}$ can be loose for real networks. The looseness of worst-case products and the threshold discontinuity noted in Sec. 2 motivate our diagnostics and allocation signals to use measured single-source impacts rather than summed analytic source predictions. Readout linearization and the finer module-source decomposition contribute additional error, retained in the exact remainder whenever Eq. 1 substitutes for the true logit error.

E.2 Non-Asymptotic Propagated-KL Certificate

Theorem 1 in Sec. 2 states the certificate for the exact logit error; this section records what the certificate's remainder term r must absorb. The bound separates approximation from consequence: Once the realized perturbation, including that remainder, is fixed, the conclusions require neither a Taylor argument nor a distributional assumption. Lemma 3 controls the state-space error made by the first-order expansion, while the readout and finer module-source errors enter r directly. The certificate is exact for the true Δz; using the propagated sources to compute Ω inherits the explicitly retained remainder r. At block-source granularity, Lemma 3 bounds the exact remainder's state-propagation part under the stated assumptions; the readout error and, at module granularity, any residual from $\xi_\ell \approx \sum_m J_{\ell,\mathrm{after}(m)}\delta_{\ell,m}$ must also be included in r. Eq. 2 both makes the Fisher remainder explicit and gives a conservative finite-perturbation companion to the two-sided Fisher comparison in Theorem 7.

E.3 Propagation-Aware Decision Radius

The preceding certificate bounds the KL of a known perturbation. Equally important for the mechanism, the inverse problem seeks the smallest perturbation, measured in the space where quantization error is injected, that can reach a decision boundary. The solution is exact for the propagated linear model.

Theorem 4 (Exact propagation-aware decision radius)**.** *At one FP prefix, let $z \in \mathbb{R}^V$ have unique winner c and pairwise margins $M_j \coloneqq z_c - z_j > 0$ for $j \ne c$. Concatenate source perturbations and the corresponding propagated maps:*

$$\delta \coloneqq (\delta_1^\top, \dots, \delta_N^\top)^\top, \qquad G \coloneqq [G_1, \dots, G_N]; \quad G\delta = \sum_a G_a\delta_a.$$

Let $\Lambda \coloneqq \mathrm{blkdiag}(\Lambda_1, \ldots, \Lambda_N) \succ 0$ *define the source norm* $\|\delta\|_\Lambda \coloneqq (\delta^\top \Lambda \delta)^{1/2}$*, and set*

$$d_j \coloneqq \left\| \Lambda^{-1/2} G^\top (e_j - e_c) \right\|_2, \qquad d_j^2 = \sum_{a=1}^{N} \left\| \Lambda_a^{-1/2} G_a^\top (e_j - e_c) \right\|_2^2, \tag{5}$$

where e_j *is the* j*th coordinate vector and* $M_j/0 \coloneqq +\infty$*. Then the exact radius for loss of the unique winner is*

$$\inf \{ \|\delta\|_\Lambda : \exists j \neq c \text{ with } (z + G\delta)_j \geq (z + G\delta)_c \} = \min_{j \neq c} \frac{M_j}{d_j}. \tag{6}$$

Each ratio M_j/d_j is the source-error norm needed to close one competitor's logit gap. The smallest ratio identifies the first decision boundary that an error can reach. A competitor with a larger FP gap can still define the nearest boundary if propagation amplifies the logit difference between the competitor and the FP winner more strongly. The radius is exact for the stated source norm and linearized map.

Least-norm boundary witness. For every j with $d_j > 0$, the least-norm perturbation that makes j tie c is

$$\delta_j^* \coloneqq \frac{M_j}{d_j^2} \Lambda^{-1} G^\top (e_j - e_c). \tag{7}$$

Corollary 5 (Exact greedy-trajectory radius)**.** *Under the setup of Theorem 4, consider an FP greedy path through horizon* H*, and suppose that one source perturbation* δ *produces the linearized logit error* $G_t\delta$ *at the FP prefix of every step* $t \leq H$*. With winner* c_t*, margins* $M_{t,j}$*, and* $d_{t,j} \coloneqq \|\Lambda^{-1/2} G_t^\top (e_j - e_{c_t})\|_2$*, the exact minimum radius for loss of the FP token as the unique decision at any of these FP prefixes, and the infimum radius for an actual greedy-token disagreement under any fixed tie-breaking rule, are both*

$$\rho_H^* = \min_{1 \leq t \leq H} \min_{j \neq c_t} \frac{M_{t,j}}{d_{t,j}}. \tag{8}$$

No independence across steps or sources is involved.

The most vulnerable token position limits the robustness of the whole continuation. If the joint source-error norm is below ρ_H^*, the FP winner remains strictly ahead of all competitors at every prefix, so the greedy paths remain identical through H. The perturbation δ may also be one joint vector that stacks step-specific sources, in which case each G_t acts on the corresponding block of δ. Taking $\Lambda = I$ gives the ordinary Euclidean source radius; a regularized local Hessian or inverse error covariance supplies a problem-adapted geometry. Eq. 5 makes the proposed mechanism exact: Decision robustness is an FP margin divided by directional propagated gain, and, under the block-diagonal source geometry, different sources' contributions to the dual directional gain d_j add in squared norm. The theorem, the boundary-witness formula, and the trajectory corollary are exact for the linearized map, and the use of all three results for the true network remains limited by the linearization remainder characterized in Lemma 3, together with readout and module-source approximation errors.

E.4 Architecture-Level Propagation Mechanisms

The factorization in Sec. 2 yields four regimes. A benign module has low fragility and low gain. A locally fragile module has high fragility but low gain, which is the regime local objectives target. An amplifier has low fragility but high gain and is therefore invisible to local objectives. A butterfly module has both high fragility and high gain. Quantization failure concentrates in the last two regimes, which depend on $G_{\ell,m}$ and are therefore missed by methods that only shrink $\|\delta_{\ell,m}\|$.

Both fragility and gain depend on the input, so the same module can remain benign on one request and strongly amplify error on another. Large amplified errors on a subset of requests form the observed divergence tail. Attention and gated MLPs can spread or amplify local quantization errors. An error in a key can shift attention toward different tokens and change how the attended tokens' values are mixed into the attention output. If several query heads share the same keys and values, an error in the shared key or value tensors affects multiple heads at once. In a gated MLP, the gate output is multiplied by the activation branch, so a large activation can magnify a small gate error. The derivations below formalize the attention-routing and gated-MLP mechanisms; Appendices I and K test the consequences of these mechanisms across architectures and KV-cache settings.

The same view identifies sites where a local perturbation can change the computation qualitatively.

Attention routing. Routing from the top key j_1 to a competitor j requires

$$\Delta S_{i,j} - \Delta S_{i,j_1} \geq S_{i,j_1} - S_{i,j}, \qquad m_i \coloneqq S_{i,j_1} - S_{i,j_2},$$

where m_i is the margin between the top two scores that the runner-up j_2 must cross. Crossing the top-two attention-score margin changes the discrete top-1 routing diagnostic, but the softmax attention map remains continuous. If $a_i \coloneqq \mathrm{softmax}(S_i)$ and only the scores change, the exact output change is

$$\Delta o_i = \sum_j (a'_{i,j} - a_{i,j}) V_j, \qquad \Delta o_i = \varepsilon(V_{j_2} - V_{j_1})$$

for a two-position transfer of mass ε from j_1 to j_2. Thus the output sensitivity is weighted by both the attention-mass change and the value contrast, not by the top-1 crossing alone.

Shared keys and values and gated MLPs. GQA and multi-query attention share keys and values across query heads, so a shared KV error propagates correlated across heads. SwiGLU has a multiplicative gate: $\Delta(g \odot a) \supset a \odot \Delta g$, so large activation branches amplify gate error. The architecture sweep in Appendix I.1 and the KV-cache study in Appendix K test these gated-MLP and routing consequences, respectively.

E.5 Exact KL Barrier for Token Changes

Theorem 4 gives the exact source-space distance to a decision boundary in the propagated linear model and shows that the controlling quantity is margin divided by directional propagated gain. The next result closes a complementary, output-distribution information-theoretic direction. Theorem 2 in the main text computes exactly how much $\mathrm{KL}(p_{\mathrm{FP}} \| p_{\mathrm{Q}})$ any top-1 change must spend. The corollary below turns the teacher-forced KL already measured in our experiments into a greedy trajectory certificate without assuming independent positions.

Corollary 6 (Teacher-forced KL certificate for a greedy trajectory)**.** *For greedy decoding under a common fixed tie-breaking rule, let p_t and q_t be respectively the FP and quantized next-token distributions evaluated under teacher forcing on the same FP prefix at step t. Assume p_t has a unique winner, put $D_t \coloneqq D_{\mathrm{KL}}(p_t \| q_t)$ and $\kappa_t \coloneqq \kappa(p_t)$, and let T be the first disagreement of the two greedy trajectories. Then, for every horizon H,*

$$[D_t < \kappa_t \text{ for all } t \leq H] \Longrightarrow T > H, \qquad \mathbf{1}\{T \leq H\} \leq \sum_{t=1}^{H} \frac{D_t}{\kappa_t}. \tag{9}$$

Consequently, for random prompts or perturbations,

$$\Pr(T \leq H) \leq \min\left\{1, \mathbb{E}\left[\sum_{t=1}^{H} \frac{D_t}{\kappa_t}\right]\right\}, \tag{10}$$

with no independence assumption across decoding steps.

Each ratio D_t/κ_t measures how much of a step's token-preservation budget the observed KL uses. A trajectory cannot first branch unless at least one step exhausts that budget, which gives the sum bound without assuming independent steps. A small FP probability gap makes κ_t small, so even modest KL can exhaust the budget at that position. The barrier covers the full vocabulary: At W3, the KL barrier remains valid when the eventual winner is no longer the FP runner-up.

E.6 Finite-Perturbation Fisher Control of the KL Tail

The exact barriers above characterize token preservation. For the magnitude of the divergence, Theorem 1 gives an additive Fisher remainder and an absolute KL ceiling. The next result instead gives exact two-sided, multiplicative control for every finite logit perturbation and then transfers that control to the complete KL distribution.

Theorem 7 (Finite-perturbation Fisher equivalence for the KL tail)**.** *Let* $p \coloneqq \mathrm{softmax}(z)$*,* $q \coloneqq \mathrm{softmax}(z+\eta)$*,* $F \coloneqq \mathrm{diag}(p) - pp^\top$*, and* $\Omega \coloneqq \mathrm{osc}(\eta)$*. Define the continuous functions*

$$c_-(\Omega) \coloneqq \begin{cases} \dfrac{2(\Omega - 1 + e^{-\Omega})}{\Omega^2}, & \Omega > 0, \\ 1, & \Omega = 0, \end{cases} \qquad c_+(\Omega) \coloneqq \begin{cases} \dfrac{2(e^{\Omega} - 1 - \Omega)}{\Omega^2}, & \Omega > 0, \\ 1, & \Omega = 0. \end{cases} \tag{11}$$

Then, with $A \coloneqq \eta^\top F \eta/2$*, the exact divergence satisfies*

$$c_-(\Omega) A \le D_{\mathrm{KL}}(p\|q) \le c_+(\Omega) A. \tag{12}$$

Both factors are strictly positive for finite Ω*, and* $c_-(\Omega) = 1 - \Omega/3 + O(\Omega^2)$ *while* $c_+(\Omega) = 1 + \Omega/3 + O(\Omega^2)$ *as* $\Omega \to 0^+$*.*

The Fisher quadratic predicts the scale of the output-distribution change from the measured logit error. Theorem 7 brackets the exact KL between two explicit multiples of that prediction for every finite perturbation. When the logit-error range is small, both factors approach one; for larger ranges, the bounds allow greater disagreement.

Corollary 8 (Fisher equivalence of the complete KL-tail distribution)**.** *Let* X *be a random token or input, and let* A_X *and* K_X *denote the Fisher quadratic and exact KL above, and suppose* $\mathrm{osc}(\eta_X) \le b$ *almost surely for some* $0 \le b < \infty$*. For every* $\alpha \in (0,1)$ *and upper-tail mass* $\tau \in (0,1]$*, let* $Q_\alpha(Y) \coloneqq \inf\{y : \Pr(Y \le y) \ge \alpha\}$ *and* $\mathrm{CVaR}_\tau(Y) \coloneqq \inf_s\{s + \tau^{-1}\mathbb{E}[(Y-s)_+]\}$*. Then*

$$\begin{aligned} c_-(b)\mathbb{E}[A_X] \le \mathbb{E}[K_X] \le c_+(b)\mathbb{E}[A_X], \\ c_-(b) Q_\alpha(A_X) \le Q_\alpha(K_X) \le c_+(b) Q_\alpha(A_X), \\ c_-(b)\,\mathrm{CVaR}_\tau(A_X) \le \mathrm{CVaR}_\tau(K_X) \le c_+(b)\,\mathrm{CVaR}_\tau(A_X). \end{aligned} \tag{13}$$

No small-error approximation or distributional-shape assumption is required beyond the stated uniform finite-range envelope.

A common finite logit-range bound lets the same two multipliers control the mean, every quantile, and CVaR of the complete KL distribution. The guarantee therefore reaches the rare, high-error inputs as well as average behavior, without assuming a particular tail shape. The lower multiplier stays positive for every finite range bound, although a large bound can make the comparison loose. The absolute certificate $\mathrm{KL} \le \Omega^2/8$ in Theorem 1 can still give a tighter worst-case ceiling; the two-sided Fisher bounds additionally control how small the KL can be relative to the quadratic.

Table 7 checks how tightly the Fisher quadratic tracks the exact KL in the evaluated regimes. We observe close empirical agreement: Plugging in the measured logit perturbation ranks the per-token KL strongly, reaching Spearman 0.91 on Qwen2.5-3B GPTQ-W4. The corresponding Fisher quadratic captures the p99 scale at a predicted-to-actual ratio of 0.86. Both findings persist across 2 independent GPTQ calibration draws (Appendix M). The under-prediction grows for the very largest perturbations that leave the local Fisher regime.

Across architectures. The empirical agreement persists across 6 pretrained architectures spanning four model families and the design grid, {RMSNorm, LayerNorm}×{GQA, MHA}×{SwiGLU, GELU, ReLU}, comprising 12 architecture×bit-width points. The parameter-free Fisher form ranks the per-token KL near-perfectly in every architecture in the near-linear W4 regime, with Spearman $\rho \ge 0.97$ and mean 0.98, and reproduces the tail magnitude at a predicted-to-actual p99 ratio of 0.89. The Fisher comparison turns conservative at W3 as the perturbations grow (Table 8).

Across scale. The agreement also survives scale: On a consistent RTN-W4 base measured identically across scale by a memory-bounded, sharding-safe two-pass forward pass, the same parameter-free quadratic ranks the deployed per-token KL at Spearman $\rho \ge 0.91$. The Fisher quadratic's mean Spearman is 0.96 across 8 model scales from 1.5B to 72B, covering the controlled Qwen2.5 ladder and cross-family Llama frontiers. The 80-layer Qwen2.5-72B reaches $\rho = 0.97$, and Llama-3.3-70B reaches $\rho = 0.91$.

Table 7: Empirical calibration of the Fisher quadratic. The evaluation uses 96 sequences of 256 tokens and 64 Colossal Clean Crawled Corpus (C4) calibration samples. Plugging the measured logit perturbation into $\Delta z_i^\top F_i \Delta z_i/2$ ranks the exact per-token KL strongly, with ρ from 0.90 to 0.96, and captures the p99 scale, although the quadratic under-predicts the p99 magnitude in these finite-perturbation cells, where the predicted-to-actual ratio runs from 0.62 to 0.92. Theorem 7 supplies the corresponding non-asymptotic two-sided guarantee.

Model	base	p99 pred.-to-actual	Spearman ρ
Qwen2.5-1.5B	GPTQ-W4	0.87	0.90
	RTN-W4	0.92	0.96
Qwen2.5-3B	GPTQ-W4	0.86	0.91
	RTN-W4	0.62	0.92

Table 8: Across-architecture Fisher check. The Fisher comparison of Table 7 extends to six architectures from four families and 12 architecture×bit points. At W4, the Fisher comparison remains near-perfect: $\rho \geq 0.97$, with mean 0.98 and mean p99 predicted-to-actual ratio 0.89. The Fisher comparison becomes conservative at W3 as perturbations grow.

Architecture	norm, attn, MLP	bits	Spearman ρ	p99 pred.-to-actual
OPT-1.3B	LayerNorm, MHA, ReLU	W3	0.93	0.86
		W4	0.99	0.98
Pythia-1.4B	LayerNorm, MHA, GELU	W3	0.88	0.74
		W4	0.98	0.95
Llama-3.2-1B	RMSNorm, GQA, SwiGLU	W3	0.74	0.77
		W4	0.97	0.87
Llama-3.2-3B		W3	0.80	0.76
		W4	0.98	0.75
Qwen2.5-1.5B		W3	0.82	0.84
		W4	0.98	0.90
Qwen2.5-3B		W3	0.84	0.80
		W4	0.98	0.92

F Proofs of the Formal Results

F.1 First-Order Remainder Bound

Write the deviation of the true error from the corresponding first-order expansion as $d_\ell := e_\ell - \sum_{k<\ell} \Phi_{k+1:\ell}\xi_k(x_k)$, with $d_0 = e_0 = 0$. From $\hat{x}_{\ell+1} = B_\ell(\hat{x}_\ell) + \xi_\ell(\hat{x}_\ell)$,

$$\begin{aligned} e_{\ell+1} &= B_\ell(x_\ell + e_\ell) - B_\ell(x_\ell) + \xi_\ell(x_\ell + e_\ell) \\ &= J_\ell(x_\ell)e_\ell + r_\ell + \xi_\ell(x_\ell) + s_\ell, \end{aligned}$$

where Taylor's theorem with integral remainder and the $L_{J,\ell}$-Lipschitz Jacobian give $\|r_\ell\| \leq (L_{J,\ell}/2)\|e_\ell\|^2$, and the mean-value bound with $\|D\xi_\ell\| \leq \kappa_\alpha$ gives $\|s_\ell\| \leq \kappa_\alpha\|e_\ell\|$. Subtracting the recursion of the first-order term, $\sum_{k<\ell+1} \Phi_{k+1:\ell+1}\xi_k = J_\ell(x_\ell)\sum_{k<\ell} \Phi_{k+1:\ell}\xi_k + \xi_\ell(x_\ell)$, yields $d_{\ell+1} = J_\ell(x_\ell)d_\ell + r_\ell + s_\ell$, hence by unrolling and $\|J_k\| \leq \Gamma_k$,

$$\|d_L\| \leq \sum_{\ell=0}^{L-1} P_{\ell+1:L}\left(\tfrac{L_{J,\ell}}{2}\|e_\ell\|^2 + \kappa_\alpha\|e_\ell\|\right).$$

The proof next bounds $\|e_\ell\|$. The same expansion gives

$$\|e_{\ell+1}\| \leq \alpha + (\Gamma_\ell + \kappa_\alpha)\|e_\ell\| + \tfrac{L_{J,\ell}}{2}\|e_\ell\|^2.$$

Because the right-hand side is nondecreasing in $\|e_\ell\|$, induction from $e_0 = \bar{E}_0 = 0$ proves $\|e_\ell\| \leq \bar{E}_\ell$. Substitution in the preceding bound for d_L proves the state-space inequality in the lemma. □

The constants $P_{a:b}$ are worst-case. Moreover, Lemma 3 starts from the block residuals ξ_ℓ; a further decomposition of each block residual into module sources introduces an additional residual, which must be retained separately whenever the finer-grained surrogate is used.

F.2 Non-Asymptotic Propagated-KL Certificate

Let $\eta \coloneqq \Delta z$. Softmax gives

$$q_j = \frac{p_j e^{\eta_j}}{\sum_k p_k e^{\eta_k}}, \qquad \log \frac{p_j}{q_j} = -\eta_j + \log \mathbb{E}_{K\sim p}[e^{\eta_K}],$$

and therefore the following identity is exact:

$$\mathrm{KL}(p\|q) = \log \mathbb{E}_{K\sim p}[e^{\eta_K}] - \mathbb{E}_{K\sim p}[\eta_K]. \tag{14}$$

First consider the Fisher approximation. Let $g(t) \coloneqq \log \mathbb{E}_{K\sim p}[e^{t\eta_K}]$. Differentiation under the finite sum gives $g'(0) = \mathbb{E}_p[\eta_K]$, $g''(0) = \mathrm{Var}_p(\eta_K) = \eta^\top F \eta$, and

$$g'''(t) = \mathbb{E}_{p_t}\left[(\eta_K - \mathbb{E}_{p_t}[\eta_K])^3\right], \qquad p_t(k) \coloneqq \frac{p_k e^{t\eta_k}}{\sum_j p_j e^{t\eta_j}}.$$

The mean under p_t lies between the smallest and largest coordinate of η, so $|g'''(t)| \le \mathrm{osc}(\eta)^3$ for every $t \in [0,1]$. Taylor's theorem applied to g between 0 and 1, together with Eq. 14, therefore yields

$$\left|\mathrm{KL}(p\|q) - \tfrac{1}{2}\eta^\top F \eta\right| \le \tfrac{1}{6}\,\mathrm{osc}(\eta)^3.$$

For the bounded moment-generating-function inequality, let a random variable Y lie in $[a,b]$ and set $\psi(t) \coloneqq \log \mathbb{E}[\exp(t(Y - \mathbb{E}[Y]))]$. Exponential tilting shows that $\psi''(t) = \mathrm{Var}_t(Y)$, the variance under the tilted law. Every law supported on $[a,b]$ obeys $\mathrm{Var}(Y) \le (b-a)^2/4$: Indeed, $(Y-a)(b-Y) \ge 0$ implies $\mathbb{E}[Y^2] \le (a+b)\mathbb{E}[Y] - ab$, hence $\mathrm{Var}(Y) \le (\mathbb{E}[Y]-a)(b-\mathbb{E}[Y]) \le (b-a)^2/4$. The tilted law has the same support, so, using $\psi(0) = \psi'(0) = 0$,

$$\psi(1) = \int_0^1 (1-t)\psi''(t)dt \le \frac{(b-a)^2}{4}\int_0^1 (1-t)dt = \frac{(b-a)^2}{8}.$$

Apply the moment-generating-function bound to $Y = \eta_K$, whose range is $\mathrm{osc}(\eta)$. Eq. 14 then yields $\mathrm{KL}(p\|q) \le \mathrm{osc}(\eta)^2/8$. The range seminorm is subadditive because

$$\mathrm{osc}(u+v) = \max_{j,k}\{(u_j - u_k) + (v_j - v_k)\} \le \mathrm{osc}(u) + \mathrm{osc}(v).$$

Applying the range-seminorm inequality repeatedly to $\eta = \sum_a v_a + r$ gives $\mathrm{osc}(\eta) \le \Omega$ and proves Eq. 2. □

F.3 Propagation-Aware Decision Radii

Proof of Theorem 4. For a competitor $j \ne c$, abbreviate $g_j \coloneqq G^\top(e_j - e_c)$. The perturbed logits make j tie or beat c exactly when

$$(z_j - z_c) + (e_j - e_c)^\top G\delta \ge 0, \qquad \text{equivalently} \qquad g_j^\top \delta \ge M_j. \tag{15}$$

By Cauchy–Schwarz after changing to the Λ geometry,

$$g_j^\top \delta = (\Lambda^{-1/2} g_j)^\top (\Lambda^{1/2}\delta) \le \|\Lambda^{-1/2} g_j\|_2 \|\delta\|_\Lambda = d_j \|\delta\|_\Lambda. \tag{16}$$

If $d_j = 0$, Eq. 15 is infeasible because $M_j > 0$, which agrees with the convention $M_j/d_j = +\infty$. If $d_j > 0$, every perturbation in that halfspace has norm at least M_j/d_j. The proposed witness satisfies

$$g_j^\top \delta_j^* = \frac{M_j}{d_j^2} g_j^\top \Lambda^{-1} g_j = M_j$$

and

$$\|\delta_j^*\|_\Lambda^2 = \frac{M_j^2}{d_j^4} g_j^\top \Lambda^{-1}\Lambda\Lambda^{-1} g_j = \frac{M_j^2}{d_j^2}.$$

The proposed witness therefore attains the lower bound and is a least-norm perturbation that makes j tie c. Loss of the unique winner is the finite union of the halfspaces in Eq. 15; taking the minimum of the halfspace distances proves Eq. 6. A strict change of winner has the same infimum, since $(1+\gamma)\delta_j^*$ lies in the strict halfspace for every $\gamma > 0$. Finally, because Λ is block diagonal,

$$d_j^2 = g_j^\top \Lambda^{-1} g_j = \sum_a (e_j - e_c)^\top G_a \Lambda_a^{-1} G_a^\top (e_j - e_c),$$

which is exactly the source decomposition in Eq. 5. □

Proof of Corollary 5. For the trajectory statement, evaluate every G_t at the FP prefix. Loss of the FP token as unique winner at one of these H prefixes is the union, over $t \le H$ and $j \neq c_t$, of the halfspaces

$$[G_t^\top (e_j - e_{c_t})]^\top \delta \ge M_{t,j}. \tag{17}$$

Before the first disagreement, the perturbed and FP decoders have the same prefix. Under any fixed tie-breaking rule, a disagreement at step t therefore implies membership in one of the closed halfspaces in Eq. 17; equality can suffice when the tie-breaking rule selects j. Conversely, for a perturbation in the corresponding open halfspace, either the trajectories have already disagreed before step t, or the perturbed and FP decoders still share the FP prefix and j strictly beats c_t at step t. Thus the disagreement set lies between the union of the open halfspaces and the union of the corresponding closures. The closures give the attained minimum for loss of uniqueness, and the open halfspaces have the same infimum by arbitrarily scaling the witness. Applying the one-step distance calculation to every pair (t, j) and taking the minimum proves Eq. 8 for either set, and hence for the disagreement set between the open and closed halfspace unions. This geometric argument makes no probabilistic assumption. □

F.4 Exact KL Barrier and Trajectory Certificate

Proof of Theorem 2. Fix a competitor $j \neq c$ and first minimize $D_{\mathrm{KL}}(p\|q)$ over the half-simplex $q_j \ge q_c$. Write $s \coloneqq p_c + p_j$ and $r \coloneqq q_c + q_j$. If some $q_k = 0$ while $p_k > 0$, the divergence is infinite and the desired bound is immediate, so the proof need only consider positive coordinates.

For fixed pair mass r, the log-sum inequality gives

$$\sum_{k \notin \{c,j\}} p_k \log \frac{p_k}{q_k} \ge (1-s) \log \frac{1-s}{1-r}, \tag{18}$$

with equality exactly when $q_k = (1-r)p_k/(1-s)$ for all $k \notin \{c, j\}$. When $s = 1$, we use the evident limiting interpretation of the equality condition. For the pair, minimizing $p_c \log(p_c/q_c) + p_j \log(p_j/q_j)$ is equivalent to maximizing $p_c \log q_c + p_j \log q_j$ subject to $q_c + q_j = r$ and $q_c \le q_j$. Without the inequality constraint, strict concavity gives $q_c = rp_c/s > r/2$ because $p_c > p_j$, so the constrained maximum lies at the boundary $q_c = q_j = r/2$. Combining the boundary optimum with Eq. 18, every feasible q satisfies

$$D_{\mathrm{KL}}(p\|q) \ge \phi_j(r) \coloneqq p_c \log \frac{2p_c}{r} + p_j \log \frac{2p_j}{r} + (1-s) \log \frac{1-s}{1-r}. \tag{19}$$

For $0 < r < 1$,

$$\phi_j'(r) = -\frac{s}{r} + \frac{1-s}{1-r} = \frac{r-s}{r(1-r)}.$$

Thus ϕ_j decreases up to $r = s$ and increases thereafter; the unique minimum of ϕ_j is attained at $r = s$. Equality in all preceding steps is attained by

$$q_c = q_j = \frac{p_c + p_j}{2}, \qquad q_k = p_k \quad (k \notin \{c, j\}),$$

and the minimum for the fixed competitor j is

$$\kappa_j \coloneqq p_c \log \frac{2p_c}{p_c + p_j} + p_j \log \frac{2p_j}{p_c + p_j}. \tag{20}$$

The event that c is not the unique maximizer is the union, over $j \neq c$, of the half-simplices $q_j \geq q_c$. The proof must identify which competitor gives the smallest κ_j. Holding p_c fixed and differentiating the right-hand side of Eq. 20 with respect to p_j yields

$$\frac{\partial \kappa_j}{\partial p_j} = \log \frac{2p_j}{p_c + p_j} < 0,$$

because $p_j < p_c$. Hence the minimum is achieved by a runner-up a, proving Eq. 4 and the asserted equality case. A zero-probability competitor is covered by continuity; in the softmax application, every coordinate is strictly positive. A strict new winner requires $q_j > q_c$, an open constraint; continuity shows that the same value is the infimum under the strict constraint. □

Proof of Corollary 6. Finally consider greedy decoding. On the event $T = t$, the two trajectories have the same prefix through step $t-1$. The two trajectories' next-token laws at the first disagreement are therefore exactly the teacher-forced p_t and q_t in the theorem, and the FP winner is not the unique winner under q_t. The result just proved implies $D_t \geq \kappa_t$. Hence $D_t < \kappa_t$ at every $t \leq H$ rules out $T \leq H$. Moreover, whenever $T \leq H$,

$$\sum_{t=1}^{H} \frac{D_t}{\kappa_t} \geq \frac{D_T}{\kappa_T} \geq 1,$$

which proves the pointwise indicator inequality in Eq. 9. Taking expectations and using $\Pr(T \leq H) \leq 1$ proves Eq. 10. No step used independence. □

F.5 Finite-Perturbation Fisher Control

Proof of Theorem 7. Fix p and η, and define the log-moment-generating function

$$h(t) \coloneqq \log \sum_{k=1}^{V} p_k e^{t\eta_k}, \qquad p_t(k) \coloneqq \frac{p_k e^{t\eta_k}}{\sum_l p_l e^{t\eta_l}}.$$

Direct differentiation of the finite sum gives

$$h'(t) = \mathbb{E}_{p_t}[\eta_k], \qquad h''(t) = \mathrm{Var}_{p_t}(\eta_k).$$

Moreover, softmax tilting gives the exact identity

$$D_{\mathrm{KL}}(p \| \operatorname{softmax}(z+\eta)) = h(1) - h(0) - h'(0) = \int_0^1 (1-t) h''(t) dt. \tag{21}$$

We first compare the tilted and untilted variances. If two probability vectors r and p obey $\alpha p_k \leq r_k \leq \beta p_k$ for every k, then, for any real vector y,

$$\mathrm{Var}_r(y_k) = \min_{a \in \mathbb{R}} \sum_k r_k (y_k - a)^2 \geq \alpha \min_{a \in \mathbb{R}} \sum_k p_k (y_k - a)^2 = \alpha \, \mathrm{Var}_p(y_k), \tag{22}$$

$$\mathrm{Var}_r(y_k) \leq \sum_k r_k (y_k - \mathbb{E}_p[y_k])^2 \leq \beta \sum_k p_k (y_k - \mathbb{E}_p[y_k])^2 = \beta \, \mathrm{Var}_p(y_k). \tag{23}$$

Let $\eta_{\min} \coloneqq \min_k \eta_k$, $\eta_{\max} \coloneqq \max_k \eta_k$, and $\Omega \coloneqq \eta_{\max} - \eta_{\min}$. Since $e^{t\eta_{\min}} \leq \sum_l p_l e^{t\eta_l} \leq e^{t\eta_{\max}}$ for $t \in [0,1]$,

$$e^{-t\Omega} p_k \leq p_t(k) \leq e^{t\Omega} p_k. \tag{24}$$

Apply Eqs. 22 and 23 with $r = p_t$ and $y = \eta$. Because $\mathrm{Var}_p(\eta_k) = \eta^\top F \eta = 2A$, we obtain

$$2Ae^{-t\Omega} \leq h''(t) \leq 2Ae^{t\Omega}. \tag{25}$$

For $\Omega > 0$, elementary integration gives

$$\int_0^1 (1-t) e^{-t\Omega} dt = \frac{\Omega - 1 + e^{-\Omega}}{\Omega^2}, \qquad \int_0^1 (1-t) e^{t\Omega} dt = \frac{e^{\Omega} - 1 - \Omega}{\Omega^2}.$$

Combining the two integral identities with Eqs. 21 and 25 proves Eq. 12. If $\Omega = 0$, all entries of η are equal; softmax is unchanged and both A and the KL are zero, so the continuous definition at zero is valid. The numerators in Eq. 11 are positive for $\Omega > 0$: The derivatives of the two numerators are respectively $1 - e^{-\Omega} > 0$ and $e^{\Omega} - 1 > 0$, and both numerators vanish at zero. Expanding the exponentials gives $c_-(\Omega) = 1 - \Omega/3 + O(\Omega^2)$ and $c_+(\Omega) = 1 + \Omega/3 + O(\Omega^2)$. □

Proof of Corollary 8. Now suppose $\mathrm{osc}(\eta_X) \le b$ almost surely. Repeating Eq. 24 with the common envelope b, rather than the realized range, yields the pointwise bound

$$c_-(b)A_X \le K_X \le c_+(b)A_X.$$

Expectation, lower quantiles, and upper-tail CVaR are all monotone and positively homogeneous for non-negative constants. For quantiles, the property follows directly from $Y \le Z \Rightarrow Q_\alpha(Y) \le Q_\alpha(Z)$ and $Q_\alpha(aY) = aQ_\alpha(Y)$ for $a \ge 0$; for CVaR, the property follows, for example, from the dual representation

$$\mathrm{CVaR}_\tau(Y) = \sup_{w \ge 0, \mathbb{E}[w]=1, \|w\|_\infty \le \tau^{-1}} \mathbb{E}[wY].$$

Applying each functional to the pointwise sandwich proves every line of Eq. 13. □

F.6 Proofs for the Propagation-Aware Low-Rank Results

Proof of Theorem 9. The equality in Eq. 27 follows by expanding the second moment and using cyclicity of the trace:

$$\begin{aligned}\mathbb{E}[\|P(E+D)h\|_2^2] &= \mathbb{E}\left[\mathrm{tr}\left(P(E+D)hh^\top(E+D)^\top P^\top\right)\right] \\ &= \mathrm{tr}\left(P(E+D)\Sigma(E+D)^\top P^\top\right) = \|P(E+D)\Sigma^{1/2}\|_F^2.\end{aligned}$$

For any feasible D, set $N \coloneqq PD\Sigma^{1/2}$. Multiplication cannot increase rank, so $\mathrm{rank}(N) \le r$, and therefore the Eckart–Young–Mirsky inequality gives

$$2\mathcal{J}(D) = \|M+N\|_F^2 \ge \min_{\mathrm{rank}(R)\le r} \|M-R\|_F^2 = \sum_{j>r} \sigma_j^2. \tag{26}$$

The Eckart–Young–Mirsky lower bound is attainable despite the left multiplication by P. Indeed, every column of M_r lies in $\mathrm{col}(M) \subseteq \mathrm{col}(P)$, and hence $PP^\dagger M_r = M_r$. With $D^* = -P^\dagger M_r \Sigma^{-1/2}$, we have

$$PD^*\Sigma^{1/2} = -PP^\dagger M_r = -M_r, \qquad \mathrm{rank}(D^*) \le \mathrm{rank}(M_r) \le r.$$

Substitution in Eq. 26 proves global optimality and Eq. 28. □

Proof of Proposition 10. We next compare with local reconstruction. Put $N \coloneqq E\Sigma^{1/2}$ and write $Z \coloneqq D\Sigma^{1/2}$. Since Σ is nonsingular, $\mathrm{rank}(Z) = \mathrm{rank}(D)$, so local rank-$r$ reconstruction solves

$$\min_{\mathrm{rank}(Z)\le r} \|N+Z\|_F^2.$$

The same truncated-SVD theorem gives $Z_{\mathrm{loc}} = -N_r$ and hence $D_{\mathrm{loc}} = -N_r\Sigma^{-1/2}$, as claimed. Let $\rho_r^2 \coloneqq \min_{\mathrm{rank}(Z)\le r} \|N+Z\|_F^2$. If P has full column rank, applying the extremal singular-value bounds of P to each column gives

$$\begin{aligned}\mathcal{J}(D_{\mathrm{loc}}) &= \tfrac{1}{2}\|P(N-N_r)\|_F^2 \le \tfrac{1}{2}\sigma_{\max}(P)^2\rho_r^2, \\ \mathcal{J}(D^*) &= \tfrac{1}{2}\min_{\mathrm{rank}(Z)\le r} \|P(N+Z)\|_F^2 \ge \tfrac{1}{2}\sigma_{\min}(P)^2\rho_r^2.\end{aligned}$$

When the optimum is nonzero, division proves Eq. 29; when $\rho_r = 0$, both the local and propagated residuals are zero and the inequality holds directly. □

Proof of Proposition 11. Take $r = 1$, $\Sigma = I$, $E = \mathrm{diag}(1+\delta, 1)$ and $P = \mathrm{diag}(1, K)$ with $K > 1+\delta > 1$. The leading singular value of E is $1+\delta$, so local truncation cancels the first diagonal entry of E and leaves $E + D_{\mathrm{loc}} = \mathrm{diag}(0,1)$. Thus $\mathcal{J}(D_{\mathrm{loc}}) = K^2/2$. On the other hand, $M = PE = \mathrm{diag}(1+\delta, K)$ has leading singular value K; the propagated optimum cancels the second entry and leaves loss $\mathcal{J}(D^*) = (1+\delta)^2/2$. The ratio of the two losses is exactly Eq. 30, and sending $K \to \infty$ rules out any propagation-independent approximation factor. □

G Training-Time Butterfly Mitigation

The decomposition $\Delta z \approx \sum_{\ell,m} G_{\ell,m}\delta_{\ell,m}$ motivates a training-time move complementary to mixed-precision selection: Instead of choosing which local errors $\delta_{\ell,m}$ to keep small, choose an effective correction according to the output consequence of the surviving error, by minimizing the propagated distributional divergence. We call the training procedure BART.

G.1 Method and Training Objective

Sec. 4.2 states the method and the objective; this section records the parameterization and the exact sense in which the KD loss is a propagated objective.

Parameterization. The correction is the LoRA product $\Delta W \coloneqq (\alpha/r)BA$ (Hu et al., 2022) with $B = 0$ at initialization, so the low-rank branch starts exactly at zero, together with a per-output-channel learnable clip ratio on the group absmax. The clips are learned quantization parameters in the OmniQuant style of Shao et al. (2024). Because the correction is folded in before quantization, the deployed artifact is the single tensor $\mathrm{quant}(W + \Delta W)$ and nothing about the inference kernel changes. Appendix G.5 gives the implementation details.

KD loss as a propagated objective. Alongside the forward KL, training carries the small NLL anchor specified in Appendix G.5. Expanding the main term at each position,

$$\mathcal{L}_{\mathrm{KD}} \coloneqq \mathbb{E}_t[\mathrm{KL}(p_t^{\mathrm{FP}} \| p_t^{\hat{\theta}})], \quad \mathrm{KL}(p_t^{\mathrm{FP}} \| p_t^{\hat{\theta}}) = \tfrac{1}{2}\Delta z_t^\top F_t \Delta z_t + R_t, \qquad |R_t| \le \mathrm{osc}(\Delta z_t)^3/6,$$

exhibits the loss as the Fisher-weighted propagated logit error up to the explicit remainder of Theorem 1: Minimizing the KD loss reduces $\|G_{\ell,m}\delta_{\ell,m}\|$, not the raw Euclidean norm $\|\Delta z_t\|_2$ and not the local $\|\delta_{\ell,m}\|$. The teacher is the same model with the correction and quantizer disabled, so no second copy is held, and cross-method cost comparisons use trained-parameter count.

G.2 Propagation-Aware Low-Rank Theory

Sufficiency of a low-rank propagated correction. The following result isolates the linear-algebraic reason the small trainable subspace can be effective. The low-rank result also separates BART's end-to-end objective from an equal-capacity local-reconstruction objective.

Theorem 9 (Optimal propagation-aware low-rank correction)**.** *Consider one linear module in a fixed-Jacobian and Fisher region. Let h be the module's random input with positive-definite second moment $\Sigma \coloneqq \mathbb{E}[hh^\top] \succ 0$, let E be the module's fixed effective weight-space error, let G propagate the module output to the logits, and let $F \succeq 0$ be the output Fisher matrix. Write $P \coloneqq F^{1/2}G$. For an effective correction D with $\mathrm{rank}(D) \le r$, define the propagated quadratic risk*

$$\mathcal{J}(D) \coloneqq \tfrac{1}{2}\mathbb{E}[\|P(E+D)h\|_2^2] = \tfrac{1}{2}\|P(E+D)\Sigma^{1/2}\|_F^2. \tag{27}$$

Let $M \coloneqq PE\Sigma^{1/2} = U\,\mathrm{diag}(\sigma_1, \sigma_2, \ldots)V^\top$ with $\sigma_1 \ge \sigma_2 \ge \cdots \ge 0$, and let M_r be a rank-r truncated SVD. Then

$$D^* = -P^\dagger M_r \Sigma^{-1/2}, \qquad \min_{\mathrm{rank}(D)\le r} \mathcal{J}(D) = \frac{1}{2}\sum_{j>r}\sigma_j^2, \tag{28}$$

where $P^\dagger$ is the Moore–Penrose pseudoinverse. In particular, $\mathrm{rank}(D^*) \le r$.

With the Jacobian and Fisher matrix fixed, a rank-r correction can cancel the r strongest modes of the module's error after propagation to the logits and Fisher weighting. The remaining loss is the energy in the modes beyond rank r, divided by two. When most propagated error energy lies in a few modes, a small trainable correction can remove most of the propagated quadratic risk.

Proposition 10 (Local reconstruction and propagation anisotropy)**.** *Under the setup of Theorem 9, for the local risk $\mathcal{J}_{\mathrm{loc}}(D) \coloneqq \mathbb{E}[\|(E+D)h\|_2^2]/2$, the rank-$r$ problem has an optimal correction $D_{\mathrm{loc}} = -(E\Sigma^{1/2})_r\Sigma^{-1/2}$ and truncates the singular modes before propagation. If P has full column rank, then*

$$\mathcal{J}(D_{\mathrm{loc}}) \le \mathrm{cond}(P)^2\,\mathcal{J}(D^*), \tag{29}$$

where $\mathrm{cond}(P) \coloneqq \sigma_{\max}(P)/\sigma_{\min}(P)$.

The condition number measures how unevenly propagation weights different error directions. If the directional gains of P are similar, local reconstruction stays close to the propagated optimum; when the directional gains differ greatly, the bound permits a large gap. If all singular values of P are equal, the condition number is one and the local correction is also optimal after propagation.

Proposition 11 (Local low-rank reconstruction can be arbitrarily suboptimal)**.** *No approximation factor independent of the propagation anisotropy is possible in Proposition 10. Indeed, for $r = 1$, $\Sigma = I$, $E = \mathrm{diag}(1+\delta, 1)$, and $P = \mathrm{diag}(1, K)$ with $K > 1+\delta > 1$, local truncation removes the first mode whereas propagated truncation removes the second mode, and*

$$\frac{\mathcal{J}(D_{\mathrm{loc}})}{\mathcal{J}(D^*)} = \frac{K^2}{(1+\delta)^2}, \tag{30}$$

which is arbitrarily large.

This two-mode example shows why the gap can grow without limit. Local reconstruction uses the single correction mode to remove the slightly larger first error, leaving the second error untouched. Propagation amplifies that second error by K, so the second error's contribution to the output loss grows as K^2. Increasing K can therefore make the local choice arbitrarily worse at the same correction rank.

We test the distinction between propagated and local objectives below against our matched-capacity OmniQuant-style local-reconstruction baseline (OMNIQ). The fixed-geometry result isolates the difference between local and propagated spectral truncation; BART directly optimizes forward KL, and the experiments evaluate the resulting correction with token-dependent Jacobians, threshold crossings, and interacting sources.

G.3 Tail-Aware Objective and Empirical Evaluation

The butterfly tail term. KD minimizes the mean propagated divergence; the largest divergences are concentrated in the upper tail. We therefore add an optional CVaR term on the per-token KL, weighted by λ, in the Rockafellar–Uryasev form (Rockafellar et al., 2000):

$$\mathrm{CVaR}_q := \min_{\nu} \nu + \tfrac{1}{q}\mathbb{E}[(\mathrm{KL}_t - \nu)_+].$$

The VaR threshold ν is a single learnable scalar optimized jointly with the weights. For fixed weights, ν is a minimizer exactly when $\Pr(\mathrm{KL}_t > \nu) \le q \le \Pr(\mathrm{KL}_t \ge \nu)$; thus the threshold is a possibly nonunique $(1-q)$ VaR. Sharing one ν across batches therefore targets the pooled per-token threshold, so the penalty falls on the upper-tail tokens across inputs rather than on the highest-divergence tokens of whichever sequence is in the current micro-batch. Our ablated per-sequence CVaR cannot see the cross-input tail and barely improves on KD.

Empirical role of the tail term. The weight λ controls the mean–tail trade-off. Across the reported W3 cells, the $\lambda = 0$ propagated-KD core provides most of the improvement. The $\lambda > 0$ variant, $\mathrm{BART}_{\mathrm{tail}}$, provides a small, model-dependent change in CVaR_1 and p99 (Table 9); we therefore treat BART as the primary variant and report $\mathrm{BART}_{\mathrm{tail}}$ as an optional upper-tail control.

BART at matched bit-widths. Table 9 compares BART and GPTQ at matched bit-widths. The effect of BART increases as the bit-width decreases.

Where BART helps. Our experiments show that the benefit grows with baseline degradation. On the low-degradation Qwen2.5 W4 cells, BART slightly worsens PPL and the KL tail, by at most 0.3 PPL and 0.2 nats of CVaR_5, so BART is recommended only where the base is degraded. Where the base is degraded, the gains are large (Table 9). At W3, BART lowers PPL from 38.12 to 12.23 on Llama-3.2-3B and from 10.24 to 7.02 on Mistral-7B, and on Qwen2.5-3B BART lowers PPL from 13.28 to 13.10 and the upper-5% per-token KL CVaR from 3.46 to 3.31. At W4 on Mistral-7B, where GPTQ's near-lossless PPL of 6.01 hides a zero-shot accuracy collapse to 0.366, BART restores accuracy to 0.683. The pattern holds across three families up to 32B.

The Mistral-7B W4 cell exhibits the butterfly effect: The average metric certifies the quantized model while the quantized-model decisions no longer match FP, and the pooled zero-shot harmful-flip rate reaches 38.8%. On the six zero-shot suites at W3 on Qwen2.5-3B, BART lifts pooled accuracy from 0.540 to 0.560 and cuts harmful prediction flips from 15.9% to 13.9%, at the same 3-bit width. The mean and KL-tail advantages grow again at W2, although zero-shot accuracy (ZS-acc) does not improve.

Table 9: BART and GPTQ at matched bit-widths. PPL, p99, and CVaR_5 and CVaR_1, the means of the worst 5% and 1% of per-token KL, use a model-specific held-out WikiText-2 sweep held fixed across methods; ZS-acc and harm instead pool the six zero-shot suites, harm being the percentage of items whose correct FP prediction becomes incorrect. The two groups therefore come from different data, so a row can pair near-lossless WikiText-2 statistics with a large zero-shot harm rate, as Mistral-7B does at W4. BART uses propagated KD with $\lambda = 0$, and $\text{BART}_{\text{tail}}$ adds CVaR with $\lambda = 0.3$.

Bits	Method	PPL	CVaR_5	CVaR_1	p99	ZS-acc	harm (%)
Qwen2.5-1.5B, FP PPL 9.28							
	GPTQ	10.00	0.77	1.56	0.97	0.593	6.1
W4	BART	10.27	0.90	1.78	1.12	0.600	5.5
	$\text{BART}_{\text{tail}}$	10.31	0.91	1.73	1.11	0.603	5.7
	GPTQ	17.23	3.98	6.40	4.93	0.483	19.1
W3	BART	15.73	3.46	5.63	4.27	0.519	15.1
	$\text{BART}_{\text{tail}}$	15.89	3.48	5.65	4.28	0.522	15.2
	GPTQ	4.71×10^6	20.91	23.81	22.19	0.301	41.7
W2	BART	1.59×10^3	11.57	13.31	12.44	0.288	39.8
	$\text{BART}_{\text{tail}}$	1.47×10^3	11.55	13.83	12.53	0.292	40.1
Qwen2.5-3B, FP PPL 8.06							
	GPTQ	8.59	0.70	1.45	0.89	0.641	3.9
W4	BART	8.83	0.90	1.75	1.13	0.632	5.5
	$\text{BART}_{\text{tail}}$	8.84	0.94	1.82	1.16	0.631	5.5
	GPTQ	13.28	3.46	5.88	4.28	0.540	15.9
W3	BART	13.10	3.31	5.81	4.16	0.560	13.9
	$\text{BART}_{\text{tail}}$	13.08	3.36	5.89	4.19	0.563	13.4
	GPTQ	3.32×10^7	24.21	27.79	25.61	0.296	44.6
W2	BART	1.49×10^3	11.90	13.74	12.65	0.289	41.8
	$\text{BART}_{\text{tail}}$	1.51×10^3	11.86	13.90	12.64	0.279	42.4
Qwen2.5-7B, FP PPL 6.72							
	GPTQ	7.10	0.60	1.20	0.73	0.666	4.4
W4	BART	7.13	0.59	1.10	0.72	0.683	3.8
	$\text{BART}_{\text{tail}}$	7.15	0.61	1.11	0.75	0.686	3.9
	GPTQ	9.56	2.64	4.56	3.27	0.612	12.5
W3	BART	9.08	2.19	3.78	2.65	0.644	9.5
	$\text{BART}_{\text{tail}}$	9.09	2.17	3.71	2.64	0.643	10.4
Qwen2.5-14B, FP PPL 4.60							
	GPTQ	7.87	4.30	6.93	5.35	0.668	10.7
W3	BART	7.16	3.44	5.55	4.29	0.693	7.8
	$\text{BART}_{\text{tail}}$	7.25	3.44	5.50	4.26	0.683	8.6
Qwen2.5-32B, FP PPL 4.51							
	GPTQ	6.55	4.39	6.91	5.48	0.695	8.5
W3	BART	5.95	3.38	5.57	4.31	0.725	6.2
	$\text{BART}_{\text{tail}}$	6.04	3.35	5.38	4.24	0.725	6.5

GSM8K recovery at W3. At W3, CoT accuracy falls from 0.847 at FP to 0.030, with 82% of problems flipped from correct to wrong (Table 11). At the same width, BART recovers accuracy to 0.350, about 41% of the FP score, and cuts the harmful-flip rate to 51%. The comparison with full-capacity QAT under the same few-hundred-example calibration regime is in Appendix G.4.

The pattern also appears in Llama-3.2-3B-Instruct: At the same 3-bit width, the GPTQ baseline drops GSM8K CoT accuracy from 0.823 to 0.277, with harmful flips at 56%, and the same lightweight BART correction recovers GSM8K accuracy to 0.420, which is about 51% of FP. The recovery is smaller than on Qwen2.5-3B-Instruct because Llama-3.2-3B-Instruct has a smaller W3 degradation to begin with. In both models, training-time shaping recovers reasoning at a fixed bit-width.

Table 9: BART and GPTQ at matched bit-widths (continued): Llama-3.2-3B and Mistral-7B.

Bits	Method	PPL	CVaR_5	CVaR_1	p99	ZS-acc	harm (%)
Llama-3.2-3B, FP PPL 8.05							
	GPTQ	10.11	2.16	5.68	3.14	0.636	4.8
W4	BART	8.71	0.68	1.49	0.84	0.635	4.4
	$\text{BART}_{\text{tail}}$	8.72	0.70	1.51	0.89	0.638	4.3
	GPTQ	38.12	7.61	12.16	9.56	0.533	16.9
W3	BART	12.23	2.81	5.13	3.64	0.578	12.4
	$\text{BART}_{\text{tail}}$	12.41	2.81	5.10	3.66	0.577	12.7
Mistral-7B, FP PPL 5.77							
	GPTQ	6.01	0.38	1.07	0.40	0.366	38.8
W4	BART	6.10	0.43	0.88	0.52	0.683	3.9
	$\text{BART}_{\text{tail}}$	6.15	0.48	1.01	0.59	0.684	4.2
	GPTQ	10.24	3.74	9.69	5.54	0.288	48.5
W3	BART	7.02	1.31	2.53	1.64	0.642	9.1
	$\text{BART}_{\text{tail}}$	7.10	1.34	2.56	1.66	0.641	9.2

Generality and scale. The bit-width pattern is not Qwen-specific. Table 9 shows the same degradation-dependent pattern: a small loss in language-model-tail metrics in W4 cells with small GPTQ degradation, improvement in high-error W4 cells, and a broader mean-and-tail win at W3 across the three model families Qwen2.5, Llama-3.2, and Mistral. The pattern also survives scale. On Qwen2.5-32B at W3, BART lowers PPL from 6.55 to 5.95 and the upper-5% per-token KL CVaR from 4.39 to 3.38 at the same 3-bit width (Table 10). Throughout, the $\lambda = 0$ propagated-KD core carries the gain and the optional tail term produces a small, model-dependent upper-tail trade-off.

Complementarity with rotation. BART is moreover complementary to the basis-change mitigations of Sec. 4: Rotation reshapes the activation geometry so the surviving local error disperses, whereas BART learns an effective correction selected by the propagated output objective. In the fixed-geometry analysis of Theorem 9, both rotation and BART alter the effective local error $E + D$; BART does not act on a mathematically separate G factor, but uses $P = F^{1/2}G$ to choose which error modes to remove. The stackability of rotation and BART is evaluated empirically. Concretely, on the Qwen2.5-3B W3 rotated GPTQ base, BART further lowers the upper-5% per-token KL CVaR from 3.37 to 3.26, showing that the two methods are complementary. Replication and paired bootstrap CIs are in Appendix G.5.

G.4 Comparison with Local and Full-Capacity Training

Matched-capacity objective comparison. To isolate the effect of the propagated objective, the local-reconstruction trainer uses the same LoRA+clip parameters, calibration data, and step budget, replacing propagated KL with the OmniQuant- and LLM-QAT-style per-block output mean-squared error (MSE). Table 10 shows lower PPL and upper-tail per-token KL CVaR for BART than for both GPTQ and the local trainer on every evaluated model. The local trainer also uses weight decay and correction clamping, as described in Appendix G.5. The local trainer's performance remains mixed relative to GPTQ and below BART on PPL and CVaR.

Comparison with full-capacity QAT. Table 10 also reports EfficientQAT, introduced by Chen et al. (2025). EfficientQAT's block-wise training of all parameters (Block-AP) trains every decoder block's entire weight together with the learnable step sizes, about 90× BART's trained parameters. The EfficientQAT learnable group-wise quantizer reduces to per-group RTN at initialization, using the same quantization primitives whose Llama-2-7B PTQ anchor lies in the published range (Table 12).

Cost–quality comparison. On Qwen2.5-3B at W3, EfficientQAT retrains the whole model and reaches PPL 10.96, compared with 13.10 for BART and 13.28 for the GPTQ-W3 base (Table 10). On GSM8K under the same few-hundred-example calibration regime (Table 11), BART exceeds LLM-QAT at 0.217 and recovers most of EfficientQAT's gain, 0.350 against 0.443, with under 2%

Table 10: W3 objective comparison. OMNIQ, BART, and $\text{BART}_{\text{tail}}$ share the same LoRA+clip capacity, data, and steps; EfficientQAT is a full-capacity block-local reference.

Objective	PPL	CVaR_5	CVaR_1	p99	ZS-acc
Qwen2.5-1.5B					
GPTQ	17.23	3.98	6.40	4.93	0.483
OMNIQ: local recon	18.04	3.86	6.27	4.66	0.472
EfficientQAT: full-capacity block-local	15.65	2.75	5.01	3.45	0.551
BART: propagated KD	15.73	3.46	5.63	4.27	0.519
$\text{BART}_{\text{tail}}$: +CVaR	15.89	3.48	5.65	4.28	0.522
Qwen2.5-3B					
GPTQ	13.28	3.46	5.88	4.28	0.540
OMNIQ: local recon	17.10	4.15	6.63	5.03	0.486
EfficientQAT: full-capacity block-local	10.96	2.37	4.48	3.01	0.583
BART: propagated KD	13.10	3.31	5.81	4.16	0.560
$\text{BART}_{\text{tail}}$: +CVaR	13.08	3.36	5.89	4.19	0.563
Qwen2.5-7B					
GPTQ	9.56	2.64	4.56	3.27	0.612
OMNIQ: local recon	9.70	2.37	4.00	2.88	0.612
EfficientQAT: full-capacity block-local	8.81	2.00	3.56	2.45	0.650
BART: propagated KD	9.08	2.19	3.78	2.65	0.644
$\text{BART}_{\text{tail}}$: +CVaR	9.09	2.17	3.71	2.64	0.643
Qwen2.5-14B					
GPTQ	7.87	4.30	6.93	5.35	0.668
OMNIQ: local recon	7.61	3.66	5.96	4.63	0.657
EfficientQAT: full-capacity block-local	7.21	3.53	5.91	4.49	0.695
BART: propagated KD	7.16	3.44	5.55	4.29	0.693
$\text{BART}_{\text{tail}}$: +CVaR	7.25	3.44	5.50	4.26	0.683
Qwen2.5-32B					
GPTQ	6.55	4.39	6.91	5.48	0.695
OMNIQ: local recon	6.16	3.63	5.97	4.56	0.685
EfficientQAT: full-capacity block-local	5.77	3.34	5.76	4.23	0.718
BART: propagated KD	5.95	3.38	5.57	4.31	0.725
$\text{BART}_{\text{tail}}$: +CVaR	6.04	3.35	5.38	4.24	0.725
Mistral-7B					
GPTQ	10.24	3.74	9.69	5.54	0.288
OMNIQ: local recon	7.22	1.62	3.39	2.04	0.642
BART: propagated KD	7.02	1.31	2.53	1.64	0.642
$\text{BART}_{\text{tail}}$: +CVaR	7.10	1.34	2.56	1.66	0.641

Table 11: GSM8K CoT recovery for Qwen2.5-3B-Instruct at W3. EfficientQAT and LLM-QAT (Liu et al., 2024a) are full-capacity QAT references.

Method	acc	flip (%)	harm (%) [95% CI]
FP	0.847	–	–
GPTQ-W3	0.030	97	82 [77,86]
EfficientQAT: W3, full-capacity	0.443	58	43 [38,49]
LLM-QAT: W3, full-capacity	0.217	78	63 [58,69]
BART: W3	0.350	65	51 [46,57]
$\text{BART}_{\text{tail}}$: W3	0.367	65	50 [44,56]

of the parameters trained by the full-capacity methods; $\text{BART}_{\text{tail}}$ recovers 82% of that gain. At 7B, $\text{BART}_{\text{tail}}$ captures 74% of EfficientQAT's upper-tail reduction: CVaR_5 falls from 2.64 to 2.17, compared with the full-capacity ceiling of 2.00, at 156× fewer trained parameters. At 32B, single-stage EfficientQAT trains 31,206M parameters against BART's 139M, a factor of 225×, and that capacity yields lower PPL, CVaR_5, and p99, whereas BART keeps higher ZS-acc and lower CVaR_1 (Table 10); the two methods trace a trained-parameter–quality trade-off rather than a strict ordering. At matched capacity, the propagated objective outperforms the local objective, as BART outperforms OMNIQ; at full capacity, EfficientQAT improves several metrics by training more parameters.

Gain-weighted full-capacity ablation. We also evaluate a full-capacity propagation-aware Block-AP variant that weights each block's reconstruction loss by a robust, unit-mean transform of the block's per-coordinate downstream gain $\|(C\Phi_{\ell+1:L})_{:,c}\|$, so capacity, compute, data, and overall loss scale match plain Block-AP. The gain-weighted Block-AP loss is a diagonal propagation proxy, not the exact propagated quadratic. Indeed, with $H_\ell \coloneqq C\Phi_{\ell+1:L}$ and block-output error e,

$$\|H_\ell e\|_2^2 = \sum_c \|(H_\ell)_{:,c}\|_2^2 e_c^2 + \sum_{c \neq c'} e_c e_{c'} \langle (H_\ell)_{:,c}, (H_\ell)_{:,c'} \rangle;$$

coordinate weighting retains a robustly compressed version of the first sum and omits the cross-coordinate terms. Thus the controlled difference is whether the full-capacity objective uses a propagation-aware diagonal surrogate or the unweighted local block-output error.

Empirically, the effect of gain weighting is real but model-dependent: On the aggressive mid-size cell, Qwen2.5-3B W3, the gain-weighted ablation cuts upper-5% CVaR from 2.37 to 2.23, a 6.1% cut. The gain-weighted ablation also changes p99 KL by 9.0% and ZS-acc from 0.583 to 0.593, but the advantage is inconsistent across sizes and bit-widths and often trades mean PPL for tail improvement, unlike the parameter-limited regime where BART's propagated objective outperforms the local objective. At full capacity, block outputs are already fit nearly exactly, leaving less capacity-limited trade-off for gain weighting to exploit.

Tail-term ablation. Comparing BART at $\lambda = 0$ with $\text{BART}_{\text{tail}}$ confirms the pattern summarized above: The KD core provides most of the benefit, while the tail term changes upper-tail metrics only slightly and inconsistently across models (Table 9).

G.5 Implementation and Optimization Details

Parameterization. BART targets attention and MLP projections while excluding tied embeddings and `Conv1D`. For every target linear module, BART freezes the FP weight W and adds a LoRA term $\Delta W \coloneqq (\alpha/r)BA$. We use rank $r = 16$ and $\alpha = 32$, initialize B to zero, and Kaiming-initialize A. Thus $\Delta W = 0$ and training starts with a zero low-rank branch. Each output channel has one learnable clip ratio applied to the channel's group-wise absmax scale. We initialize the clip ratio to $\sigma^{-1}(0.982)$, near full absmax. The deployed artifact is a single symmetric quantized tensor $\text{quant}(W + \Delta W)$ with group size $g = 128$ at the PTQ bit-width. The correction is folded in before quantization, so the model has the same storage size and uses the same inference kernel as the GPTQ baseline. We evaluate both under the same protocol. Trained parameters are under 2% of the model. Because every matrix of rank at most r admits a factorization BA, the LoRA parameterization spans exactly the effective correction class analyzed in Theorem 9 before the quantizer is re-applied. The actual-KL training run then evaluates threshold crossings and input-dependent propagation directly.

Objective and optimization. The student uses quantized, straight-through rounding and is distilled from the teacher, which uses the same weights with the correction and quantizer disabled. The objective and tail functional are

$$\mathcal{L} \coloneqq \mathcal{L}_{\text{KD}} + \lambda \mathcal{L}_{\text{CVaR}_q} + \beta_{\text{NLL}} \mathcal{L}_{\text{NLL}}, \qquad \text{CVaR}_q \coloneqq \nu + \tfrac{1}{q}\mathbb{E}[(\text{KL}_t - \nu)_+].$$

Here $\mathcal{L}_{\text{KD}}$ is mean forward token-KL to the teacher, $\beta_{\text{NLL}} = 0.1$ is a small NLL anchor, and the tail term is Rockafellar–Uryasev CVaR at $q = 0.05$. The CVaR threshold ν is a single scalar trained jointly with the weights and targets pooled VaR across steps. We warm-start the threshold at the first batch's empirical $(1 - q)$-quantile. We use adaptive moment estimation (Adam) with decoupled weight decay (AdamW), using $(\beta_1, \beta_2) = (0.9, 0.95)$, LoRA learning rate 2×10^{-4}, and a learning

rate for the clip and ν that is five times as large. Training uses a cosine schedule, gradient-norm clip 1.0, 500 steps of one 512-token C4 sequence each from a 256-sequence calibration pool, and gradient checkpointing. BART uses $\lambda = 0$ as the KD core, whereas $\text{BART}_{\text{tail}}$ uses $\lambda = 0.3$; raising λ traces the mean–upper-tail trade-off. A within-micro-batch top-k CVaR ablation cannot see the cross-input tail and barely changes CVaR_1 or p99 relative to KD.

Cost, replication, and CIs. Trained-parameter count is the cross-method cost axis. Across three independent calibration and training runs, the 1.5B-W3 CVaR_5 gap between BART and GPTQ is stable, with paired bootstrap 95% CI -0.498 $[-0.563, -0.434]$.

The matched-capacity local baseline, OMNIQ. The local-reconstruction trainer in Table 10 shares BART's parameterization exactly: rank-16 LoRA, a per-output-channel learnable clip, the same calibration data, the same step budget, and the same AdamW optimizer family, betas, and schedule. The local-reconstruction trainer differs only in the objective, minimizing the OmniQuant-style summed relative MSE between quantized and FP block hidden-state outputs rather than the propagated token-KL. The local-reconstruction trainer also uses weight decay of 10^{-3} and a per-step projection that clamps each channel's correction to $0.2\times$ the corresponding base-row norm and floors the learnable clip ratio. Both measures stabilize the otherwise-divergent free local objective on the deepest models.

Reference implementation. Listing 1 gives the paper-facing PyTorch core of BART. The listing retains the operations that define the method and omits architecture traversal, gradient checkpointing, sharded execution, and the final deployment-time weight fold. The retained operations are the frozen FP teacher weight, the low-rank correction, learnable group clipping, straight-through quantization, forward token-KL, and the single shared VaR threshold. Setting `tail_weight=0` gives the BART core; setting `tail_weight` to 0.3 gives $\text{BART}_{\text{tail}}$. The objective module's parameter ν is included in the optimizer together with the LoRA and clip parameters.

```python
import math
import torch
import torch.nn as nn
import torch.nn.functional as F

def ste_group_quant(weight, bits, group_size, clip_logit):
    """Symmetric group quantization; backward uses a straight-through estimator."""
    out_features, in_features = weight.shape
    group = group_size if in_features % group_size == 0 else in_features
    weight_g = weight.reshape(out_features, in_features // group, group)
    qmax = 2 ** (bits - 1) - 1
    absmax = weight_g.abs().amax(dim=-1, keepdim=True).clamp_min(1e-8)
    scale = torch.sigmoid(clip_logit).reshape(out_features, 1, 1)
    scale = scale * absmax / qmax
    scaled = weight_g / scale
    levels = scaled + (scaled.round() - scaled).detach()
    levels = levels.clamp(-qmax - 1, qmax)
    return (levels * scale).reshape(out_features, in_features)

class BARTLinear(nn.Module):
    """Frozen FP linear plus a quantization-aware low-rank correction."""
    def __init__(self, base, bits, group_size=128, rank=16, alpha=32.0):
        super().__init__()
        self.base = base
        for parameter in self.base.parameters():
            parameter.requires_grad_(False)
        out_features, in_features = base.weight.shape
        device = base.weight.device
```

```
        self.A = nn.Parameter(torch.empty(rank, in_features,
    ↪ device=device))
        self.B = nn.Parameter(torch.zeros(out_features, rank,
    ↪ device=device))
        nn.init.kaiming_uniform_(self.A, a=math.sqrt(5))
        self.clip_logit = nn.Parameter(
            torch.full((out_features, 1), 4.0, device=device))
        self.bits, self.group_size = bits, group_size
        self.lora_scale = alpha / rank
        self.student = True

    def forward(self, inputs):
        if not self.student:                    # FP teacher: W only
            return F.linear(inputs, self.base.weight, self.base.bias)
        delta = (self.B @ self.A).to(self.base.weight.dtype)
        effective = self.base.weight + self.lora_scale * delta
        quantized = ste_group_quant(
            effective.float(), self.bits, self.group_size,
    ↪ self.clip_logit)
        return F.linear(inputs, quantized.to(inputs.dtype),
    ↪ self.base.bias)

class BARTObjective(nn.Module):
    """Mean forward KL + shared-threshold CVaR + next-token NLL."""
    def __init__(self, tail_weight=0.0, tail_q=0.05, nll_weight=0.1):
        super().__init__()
        self.tail_weight, self.tail_q = tail_weight, tail_q
        self.nll_weight = nll_weight
        self.nu = nn.Parameter(torch.zeros(()))
        self._nu_initialized = False

    def forward(self, teacher_logits, student_logits, tokens):
        teacher_logp = F.log_softmax(teacher_logits.float(),
    ↪ dim=-1).detach()
        student_logp = F.log_softmax(student_logits.float(), dim=-1)
        token_kl = (teacher_logp.exp()
                    * (teacher_logp - student_logp)).sum(dim=-1)

        if not self._nu_initialized:          # first-batch (1-q)-VaR
    ↪ warm start
            with torch.no_grad():
                k = max(1, int(round(token_kl.numel() * self.tail_q)))
                threshold = torch.topk(token_kl.reshape(-1),
    ↪ k).values.min()
                self.nu.copy_(threshold)
            self._nu_initialized = True

        kd = token_kl.mean()
        cvar = self.nu + F.relu(token_kl - self.nu).mean() /
    ↪ self.tail_q
        vocab = student_logits.shape[-1]
        nll = F.cross_entropy(
            student_logits[:, :-1].reshape(-1, vocab).float(),
            tokens[:, 1:].reshape(-1))
        loss = kd + self.tail_weight * cvar + self.nll_weight * nll
        return loss, token_kl, cvar
```

Listing 1: `BARTLinear` and the pooled-CVaR objective. The shared learnable threshold `nu` (ν) persists across optimization steps.

H Experimental Details

H.1 Models, Quantizers, Data, and Metrics

Models. The Qwen2.5 ladder (Yang et al., 2024a) comprises the 1.5B, 3B, 7B, 14B, 32B, and 72B models and uses RMSNorm, rotary position embeddings (RoPE) (Su et al., 2024), GQA (Ainslie et al., 2023), and SwiGLU (Shazeer, 2020). Llama-3.2-1B, Llama-3.2-3B, Llama-3.1-8B, and Llama-3.3-70B (Team, 2024) and Mistral-7B (Jiang et al., 2023) share this RMSNorm, RoPE, GQA, and SwiGLU design; Llama-2-7B (Touvron et al., 2023), which uses MHA in place of GQA, serves as the standard validation cell (Table 12). The LayerNorm and MHA side of the grid is covered by Pythia-1.4B, which is based on GPT-NeoX and uses RoPE, GELU, and a parallel-residual design (Biderman et al., 2023); by OPT-1.3B, which uses learned positional embeddings and ReLU (Zhang et al., 2022); and by GPT-2-large, which uses learned positional embeddings and GELU (Radford et al., 2019). Depths range from 16 layers in Llama-3.2-1B to 80 layers in the 70B and 72B models. Instruction-tuned checkpoints, marked `-Instruct`, are used for the generative and deployment benchmarks. Every number we report comes from the checkpoint named in the corresponding table, figure, or caption, and the heavier diagnostic, causal, and allocation studies run on the $\leq$32B subset. Unless stated otherwise, the FP reference uses bf16.

Quantizers. We use five weight-only PTQ bases. The two primary bases are RTN and an implementation of GPTQ (Frantar et al., 2023); GPTQ uses group size 128 and the standard C4 calibration of 128×2048 tokens. The third is an AWQ-style activation-aware base whose per-channel scale search minimizes true local output error. The fourth, a QuaRot-style residual-basis base, folds norm scales into adjacent linears, applies a randomized Hadamard rotation to the residual basis, which is the transform QuaRot uses, and then runs RTN or GPTQ in the rotated coordinates. The fifth, a learned-rotation base, follows SpinQuant: The base uses a Cayley-parameterized orthogonal rotation initialized from the Hadamard base and trains the rotation to minimize quantization error in the rotated weights. The AWQ-style and learned-rotation bases are detailed in Appendices H.2 and H.3. All quantizers are weight-only at 3 or 4 bits unless stated otherwise. We use the Wb notation defined in Sec. 3. The residual-basis implementation is evaluated at group size 128. Mitigation experiments use the stated base quantizer, with the same average bit-width budget across allocation scores and the same weight bit-width across training methods. Table 12 in Appendix H.2 validates the implementation on the standard Llama-2-7B group-128 cell: At W3, our GPTQ value differs from the published value by 0.08 PPL, RTN matches the published value, and AWQ is 0.25 PPL higher under a different calibration corpus.

Data. We use WikiText-2 (Merity et al., 2017) for PPL with 2048-token windows and for per-input divergence; C4 (Raffel et al., 2020) for calibration; and six standard zero-shot suites scored by length-normalized log-likelihood. The suites are HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), the Easy and Challenge splits of ARC (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2020), and LAMBADA (Paperno et al., 2016). Generation prompts come from WikiText-2 and Dolly-15k instructions. We use GSM8K (Cobbe et al., 2021) for generative CoT evaluation with Qwen2.5-3B-Instruct and, for the cross-family BART check, Llama-3.2-3B-Instruct.

Metrics. We report the mean PPL gap; the per-token $D_{\mathrm{KL}}(p_{\mathrm{FP}} \| p_{\mathrm{Q}})$ distribution between FP and quantized outputs, including the distribution's mean, the 95th percentile, p99, and p99.9; top-1 flip rate; greedy generation divergence, defined as the fraction of prompts whose greedy continuation differs within 48 tokens; and on downstream tasks the per-example prediction-flip and harmful-flip rates between the FP and quantized model. Experiments used a node with four A100-40GB graphics processing units (GPUs). Large models were sharded as needed; the 7B allocation study used one GPU with layer-chunked, host-resident Hessians.

H.2 Evaluation and Quantization Protocols

Divergence evaluation. The primary phenomenon and localization evaluations use 160 to 256 non-overlapping 512-token WikiText-2 windows per model; smaller diagnostic probes state sample counts in the associated captions. Per-token KL is $D_{\mathrm{KL}}(p_{\mathrm{FP}} \| p_{\mathrm{Q}})$ at each position under teacher forcing on the same input.

Generation and null control. Generation divergence in Table 5 for W4 uses greedy decoding for 48 new tokens from 24 to 32 held-out prompts. The null-controlled sweep in Table 6, which also supplies the W3 entries, uses 384 prompts per model: 256 32-token WikiText-2 prefixes and 128 Alpaca-formatted Dolly-15k instructions (Conover et al., 2023). Null conditions execute the identical FP weights in three ways: generation inside left-padded batches of 8, the eager attention kernel in place of PyTorch SDPA, or an fp16-cast copy of the weights. The teacher-forced batch null forwards the same sequence as row 0 of an 8-row batch. We compute PPL using the standard GPTQ protocol with 2048-token windows. All comparisons between FP and quantized models for a given experiment run under identical attention-mask code paths and identical batch composition.

Allocation budgets. Mixed precision uses two-level allocation: Every signal ranks modules, and a module is promoted greedily unless doing so would exceed the common parameter-weighted average bit-width target. Thus every signal operates under the same capacity constraint and never exceeds the target, although indivisible, heterogeneous modules can leave different small amounts of unused budget. The held-out hard subset is the top KL decile of the evaluation set under the uniform base quantizer, fixed before comparing signals.

Butterfly-impact signal. The single-source sweep quantizes one module at a time at the lower bit-width and measures the mean per-token output KL on the hard decile of a held-out calibration scan using 48 $\times$ 512-token C4 windows. Cost is one forward per module per calibration sequence, comparable to GPTQ's Hessian collection.

KV-cache simulation. Quantization is applied to the post-RoPE, pre-replication key and value states at the attention interface, which is exactly what a quantized cache stores under GQA. Per-channel K scales span 32-token blocks; in streaming decode, the trailing partial block stays FP as the KIVI residual, while in teacher-forced scoring all full blocks are quantized. Routing-flip statistics compare $\arg\max_j S_{ij}$ before and after K quantization per query-head pair under the causal mask on 32 evaluation sequences; relative errors are Frobenius norms of the key and value perturbations.

AWQ-style base. The second strong base searches a per-input-channel scale $s_j \coloneqq \mathrm{rms}(x_j)^\alpha$. The activation root-mean-square magnitude in s_j comes from the same calibration second moments as GPTQ's Hessians. For each module, α is selected on an 11-point grid to minimize the true local output error $\mathrm{tr}(\Delta W H \Delta W^\top)$. The method then quantizes $Q(W\mathrm{diag}(s))\mathrm{diag}(s)^{-1}$, the exact weight-only equivalent of AWQ's folded transform (Lin et al., 2024); clipping search is omitted.

GSM8K protocol. We use Qwen2.5-3B-Instruct with the Qwen2.5-3B-Instruct chat template and a zero-shot CoT prompt that instructs the model to reason sequentially and place the final answer within `boxed{}`. We decode greedily to 512 tokens in fixed batches of 16 with identical batch composition across FP and all quantized conditions, over 300 test problems. Answers are parsed from the last `boxed{}`; if the boxed answer is absent, we use the last number. We compare answers numerically with the gold answer and the FP answer.

Statistical reporting. Pooled task rates carry 95% bootstrap CIs using 10,000 resamples over examples. Causal-recovery percentages use 95% bootstrap CIs over evaluation sequences with 2,000 resamples. KV-setting p99 sequence-KL values use cluster bootstrap intervals over 2,000 resampled sequences. The weight-mitigation comparison is additionally repeated over three independent GPTQ calibration draws, with calibration windows resampled and evaluation data fixed. The evaluation protocol includes a paired sequence-bootstrap CI of the Hessian−impact hard-subset difference; Appendix M gives the details.

Implementation validation. Table 12 compares the RTN, GPTQ, and AWQ-style implementations with published WikiText-2 PPL values on the standard Llama-2-7B group-128 cell and records the two rotation bases on the same cell.

H.3 Rotation Construction

Rotation base. The rotation preprocessing first folds every RMSNorm scale γ into the adjacent linear maps. For input-reading matrices, the norm fold gives $W \leftarrow W\mathrm{diag}(\gamma)$; the final norm

Table 12: WikiText-2 PPL for Llama-2-7B weight-only quantization at group size 128 and sequence length 2048. Published values are from Shao et al. (2024); rotation rows use the randomized Hadamard residual basis of Ashkboos et al. (2024). Rotation followed by RTN collapses at W3 on Llama-2-7B, so the rotated bases of Sec. 4 pair rotation with GPTQ.

	W4		W3	
Method	ours	pub.	ours	pub.
FP (bf16)	5.47, matching the published value			
RTN	5.73	5.72	6.66	6.66
GPTQ	5.63	5.61	6.37	6.29
AWQ	5.62	5.62	6.49	6.24
OmniQuant	–	5.58	–	6.03
Rotation+RTN	7.29	–	4×10^3	–
Rotation+GPTQ	5.73	–	8.90	–

folds into the language-model head and is untied from the embedding. The procedure then inserts a randomized Hadamard matrix Q into the residual basis: Input-reading matrices use $W \leftarrow WQ$, residual-writing matrices use $U \leftarrow Q^\top U$ with the same left rotation applied to the corresponding biases, and embeddings $E \leftarrow EQ$. Q is the orthonormal Walsh–Hadamard matrix constructed by the Sylvester method. For a non-power-of-two hidden size $d = 2^a m$, we use $H_{2^a} \otimes Q_m$, where Q_m is Haar-random orthogonal on the odd remainder. This residual-basis transform is post-multiplied by a random sign diagonal, the residual-basis transform used by QuaRot (Ashkboos et al., 2024). In exact arithmetic, unit-scale RMSNorm is rotation-equivariant and the reparameterized model computes the same function. We perform all folds and rotations in 64-bit floating-point arithmetic, store the result in bf16, and observe PPL differences of $+0.08\%$ on Llama-2-7B and $+0.04\%$ on Qwen2.5-3B, consistent with bf16 storage round-off. Quantization experiments operate at group size 128; Hessians, gains, and impact sweeps for the rotated allocation study are collected on the rotated model unchanged.

Rotation configuration. The study uses the residual-basis rotation and excludes the in-head rotation and online activation transforms. The SpinQuant-style learned-rotation variant starts from the Hadamard Q_0 and learns a Cayley-parameterized correction $Q := (I - S)(I + S)^{-1}Q_0$, where S is skew-symmetric, with Adam against the straight-through group-quant MSE. The initialization $S = 0$ recovers the Hadamard base exactly, making the Hadamard base an attainable reference. We report the learned-rotation result alongside the Hadamard reference.

H.4 Diagnostic Proxies, Module Scores, and Matched-Input Audit

This section records the definitions and caveats behind the diagnostics of Sec. 2.3: the single-forward computation of the per-block local perturbation, the relation between the implemented gain proxy and the approximated operator, the auxiliary module-level scores reported in the allocation tables, the input-level index, and the matched-input audit.

Per-block local perturbation. On the FP trajectory $\{x_\ell\}$, we compute

$$\xi_\ell(x_\ell) = \hat{B}_\ell(x_\ell) - B_\ell(x_\ell) = \hat{B}_\ell(x_\ell) - x_{\ell+1}$$

in a single forward by overriding every block's input with the block's FP residual state and reading the quantized block output.

Downstream gain. The gain exactly paired with the block-output residual ξ_ℓ in the linearized recurrence is the post-block operator $C\Phi_{\ell+1:L}$, which can be probed by injecting immediately after block ℓ, at residual state $x_{\ell+1}$: $x_{\ell+1}$ is the next block's input or the readout input after the final block. Our implemented layer-level diagnostic instead injects unit-norm noise at the input of block ℓ and measures $\|\Delta z\|/\|\text{noise}\|$, averaged over random directions in the model's native output gauge. We denote the deliberately coarser, quantization-free proxy by $\widetilde{G}_\ell$; to first order the proxy probes $C\Phi_{\ell:L}$

and therefore includes the Jacobian of block ℓ in addition to the downstream map paired with ξ_ℓ. Replacing Δz by $\Pi_\perp \Delta z$, where $\Pi_\perp := I - \mathbf{1}\mathbf{1}^\top/V$ projects away the constant-logit direction, gives the corresponding canonical softmax-invariant proxy.

Layer-selection score. For the butterfly score $b_\ell := \widetilde{G}_\ell \cdot \mathbb{E}_x[\|\xi_\ell(x)\|]$ of Sec. 2.3, equality with $\|C\Phi_{\ell+1:L}\xi_\ell\|$ is not asserted: The probe averages over directions and is taken one block before the injection site of ξ_ℓ.

Module-allocation scores. We use distinct names for the three diagnostic granularities: The butterfly score ranks layers for causal restoration, butterfly-impact ranks modules for PAPA, and QBI identifies high-risk inputs. At module level, let $H_m := \mathbb{E}[h_m h_m^\top]$ denote the calibrated input second moment, which is the usual linear-layer Hessian proxy. The quantity

$$e_{\ell,m} := \mathrm{tr}(\Delta W_m H_m \Delta W_m^\top) = \mathbb{E}[\|\Delta W_m h_m\|_2^2]$$

is the local reconstruction energy. The allocation tables include the auxiliary heuristic $\widetilde{G}_\ell e_{\ell,m}$ under the name gain-weighted local proxy; the heuristic is the implemented ranking score, not a dimensionally exact propagated quantity. Under an isotropic scalar-proxy approximation, the corresponding propagated norm and energy would instead scale as $\widetilde{G}_\ell \sqrt{e_{\ell,m}}$ and $\widetilde{G}_\ell^2 e_{\ell,m}$, respectively. PAPA uses the measured single-source impact

$$I_{\ell,m} := \mathbb{E}_{x \in \mathrm{hard}}[D_{\mathrm{KL}}],$$

which Sec. 2.3 introduces as butterfly-impact and Sec. 4 uses as the allocation signal.

QBI. $\mathrm{QBI}(x) := \overline{D}_{\mathrm{KL}}(p_{\mathrm{FP}} \| p_{\mathrm{Q}})/(E_{\mathrm{loc}}(x) + \epsilon)$, where $\overline{D}_{\mathrm{KL}}$ is the mean over the evaluated token positions, $\epsilon > 0$, and E_{loc} is the mean per-block local energy: Large QBI marks inputs whose output changes disproportionately to the input's local error. The QBI serves as an input-level diagnostic. The hard subset used by the mitigation is instead the top decile of per-sequence KL under the uniform base quantizer, fixed before any allocation rule is compared (Appendix H.2); the hard subset is not selected by QBI.

Matched-input audit. Figure 4 makes the distinction between local perturbation magnitude and propagated effect concrete, and records how the Spearman $\rho = 0.012$ reported in Sec. 2.3 was obtained over the 256 held-out WikiText-2 windows. For the qualitative comparison, we apply a deterministic extremal selector: Among the 703 input pairs whose local energies differ by at most 0.5%, we take the pair with the largest output-KL ratio. The selected inputs differ by only 0.11% in local energy but by $4.27\times$ in output KL. The population scatter, not the selected pair's extremal ratio, carries the statistical claim; the pair is a concrete witness to what the scatter implies.

I Extended Empirical Validation

I.1 Architecture and Distribution Dependence

Table 13 runs the diagnostics across five real families spanning all three design axes.

Architecture dependence. Causal concentration is an architectural property: Butterfly selection gives the largest recovery on Qwen2.5-3B and Llama-3.2-1B, statistically ties Hessian on Qwen2.5-1.5B, and no tested ranking beats random on Pythia-1.4B (Table 1). More broadly, the upper-tail and prediction-flip signature appears in every evaluated family, while the dominant hotspot shifts from the SwiGLU `down` projection to attention `qkv` or the first MLP projection. The per-model analysis and the complete architecture and cross-distribution sweep follow below.

Across the evaluated families. We observe the same signature in every evaluated family: a pronounced upper tail of per-token KL at 5.7 to 7.2 times the mean, together with a flip rate of 13% to 25% at RTN-W4.

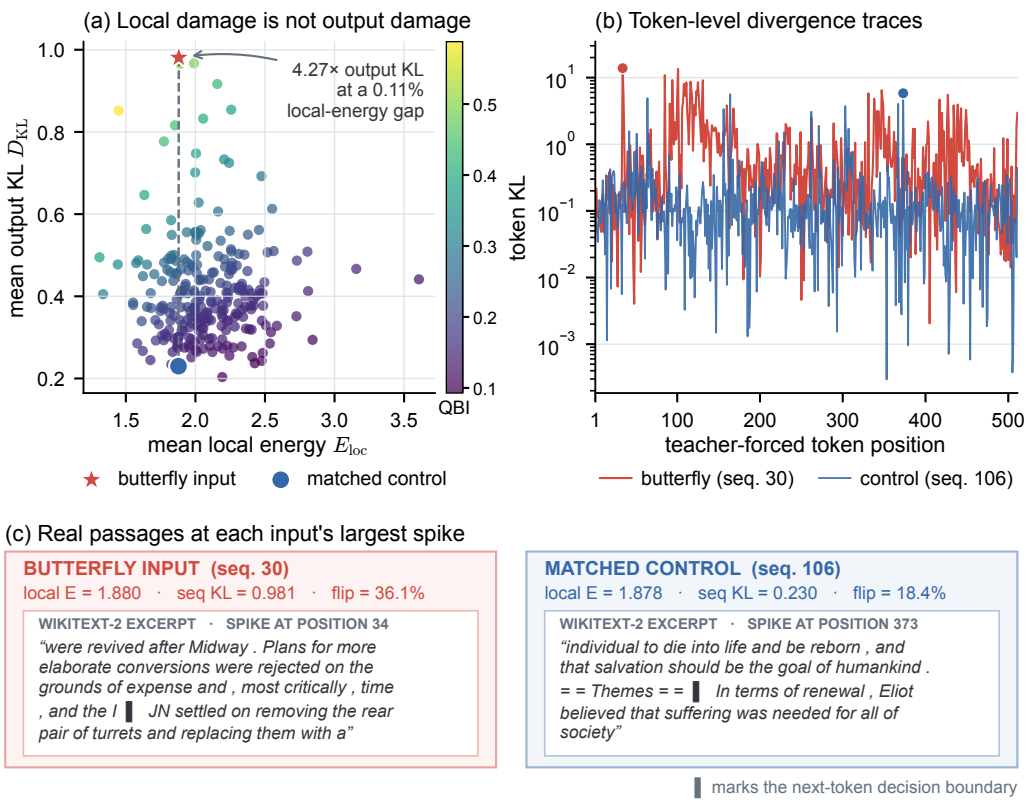


Figure 4: Matched-input audit showing that local error alone does not determine propagated failure. On 256 held-out WikiText-2 windows from Qwen2.5-3B under RTN-W4, the scatter in (a) covers the population together with a pair matched to within 0.5% local energy. The selected inputs' mean output KL differs by $4.27\times$, and (b) and (c) resolve the selected pair into within-sequence tails and passages. The population scatter, rather than the selected pair alone, supports the conclusion.

Hotspot location. The dominant single-source module role tracks the architecture: The hotspot is the SwiGLU `down` projection in Qwen2.5 and Llama-3.2, but shifts to the attention `qkv` or first MLP projection in Pythia, OPT, and GPT-2. The shift in hotspot location is consistent with the multiplicative-gate and attention-routing mechanisms of Sec. 2.

Architecture-specific concentration. The per-model results of Table 1 expose where the causal concentration of Sec. 3.2 does and does not hold. On Qwen2.5-3B the butterfly score gives the largest recovery, 83.1 [82.7,83.4] against 67.8 [55.0,77.5] for Hessian. On Qwen2.5-1.5B, the top selections statistically tie, at 45.6 [43.6,48.5] for butterfly and 48.8 [41.3,50.6] for Hessian. Llama-3.2-1B has 16 layers, so $k = 4$ protects a quarter of the network. There the methods differ substantially: Butterfly recovers 55%, compared with 20% for Hessian and 15% for local error; there the local signals do worse than random, at 41%, because the local signals prioritize late high-reconstruction-error layers that the tail does not pass through. In Pythia-1.4B, no tested selection significantly beats random, with butterfly at 28% and random at 30%, while local and Hessian still trail at 14%. Where causal concentration holds, the gain-aware score finds the concentrated layers; where causal concentration does not hold, as in the Pythia-1.4B RTN-W3 restoration experiment, none of the tested rankings provides a clear advantage over random selection, and the practical benefit from selective protection, ours included, is set by the design.

Across data distributions. Our held-out C4 evaluation shows that the statistics are not corpus-specific. The same GPTQ-W4 models keep a $9.0\times$ per-token KL tail, 14.9% flips and 98% generation divergence behind a 7.8% mean PPL gap, averaged over Qwen2.5-1.5B and Qwen2.5-3B. The disproportionate-tail structure even crosses the distribution gap between natural language and code.

Table 13: Architecture sweep under RTN-W4. All five families exhibit the tail and prediction-flip signature; the dominant hotspot shifts from the SwiGLU `down` projection in gated-MLP models to attention `qkv` or the first MLP projection in MHA models.

Model	Norm	Attn	MLP	tail (×)	flip (%)	hotspot
GPT-2-large	LayerNorm	MHA	GELU	6.6	12.9	`up`
OPT-1.3B			ReLU	6.6	21.8	
Pythia-1.4B			GELU	5.7	19.3	`qkv`
Llama-3.2-1B	RMSNorm	GQA	SwiGLU	7.1	25.3	`down`
Qwen2.5-1.5B				7.2	21.8	

On a held-out code corpus, CodeParrot, GPTQ-W4 on Qwen2.5-3B leaves a per-token KL tail 9.7× the mean with 11% of tokens flipping. The code-corpus result has the same upper-tail signature, now on inputs such as a variable name or a constant where a single flip is most consequential. The phenomenon also holds on a second non-Qwen family quantized with plain GPTQ: On Mistral-7B, a 3.9% mean PPL gap hides a 9.0× per-token KL tail, 9% flips and 94% generation divergence (Table 5).

At the input level, a learned probe over FP propagation features ranks high-risk inputs and enables input-adaptive routing as described in Appendix N; module localization and input prediction use distinct operational scores.

I.2 Tail Failure on Standard Deployment Benchmarks

Broader deployment benchmarks. The same per-item audit extends to Massive Multitask Language Understanding (MMLU), execution-scored HumanEval and Mostly Basic Python Problems (MBPP), and Instruction-Following Evaluation (IFEval): The median GPTQ-W4 accuracy shift is 2.7 points, while harmful flips reach 30% in the worst cell. The complete benchmark matrix, execution protocol, CIs, and the figure contrasting mean shift with harmful-flip rate follow in this subsection.

The flip tail above was measured on six zero-shot multiple-choice suites and GSM8K. We next test whether the flip tail persists on standard deployment benchmarks not covered in the earlier sections. We extend the same per-item audit between FP and quantized models, comparing each quantized model to the corresponding FP predictions example by example, to MMLU 5-shot (Hendrycks et al., 2021), to HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021) scored by functional correctness. We use pass@1, where each generated program executes against the benchmark's own unit tests in a sandboxed subprocess. We also evaluate IFEval (Zhou et al., 2023) instruction following across 5 instruction-tuned models (Tables 14 and 15).

The bases are the same GPTQ and RTN implementations whose standard-cell PPL values lie in the published OmniQuant and AWQ range. GPTQ is within 0.08 PPL, as Table 12 shows, and the FP baselines solve the majority of every benchmark.

Figure 5 shows exactly the signature that the average hides. Under the strong GPTQ-W4 base, a typical cell barely moves. The median $|\Delta\mathrm{acc}|$ across cells is 2.7 points. The small median accuracy shift is not universal, however. The worst HumanEval cell drops 24.4 points.

Per-item failures on deployment benchmarks. In either case, a large fraction of individual items flip, of which a harmful share regresses a correct answer to a wrong answer: up to 30% on the worst cell.

Code and instruction following. Code is the most exposed axis, because functional correctness is brittle to a single wrong token. On Qwen2.5-3B-Instruct, HumanEval pass@1 drops from 0.750 to 0.506, while 35% of problems flip outcome and 30% regress from pass to fail (Table 14). Instruction following shows the same pattern. Mean instruction-following accuracy moves by at most five points and even rises slightly on one model, while 9% to 13% of instructions regress from pass to fail underneath; beneficial flips mask part of the harmful flips in the average. At the aggressive W3

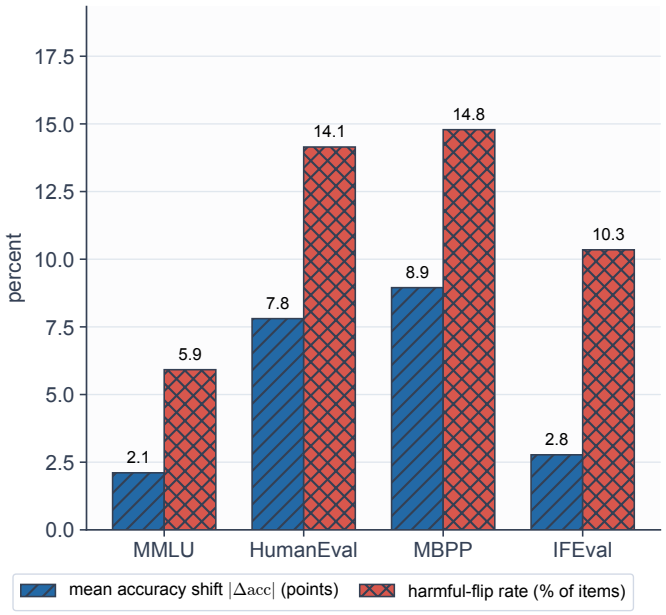


Figure 5: The mean hides the tail on standard deployment benchmarks. Under the strong GPTQ-W4 base, the benchmark-averaged accuracy shift $|\Delta\mathrm{acc}|$, drawn as forward-hatched bars, is smaller than the per-item harmful-flip rate, drawn as cross-hatched bars, on knowledge in MMLU, code scored by execution in HumanEval and MBPP, and instruction following in IFEval. The cell-level mean shift is not uniformly small: The worst HumanEval cell drops 24.4 points. A typical cell can show little aggregate degradation at 4 bits without matching the deployed FP model.

Table 14: Per-item reliability on standard deployment benchmarks under GPTQ-W4. The $|\Delta|$ column is the unsigned accuracy shift in percentage points, so the column does not distinguish a drop from a gain; IFEval on Llama-3.2-3B-Instruct is the one cell where accuracy rises. The harm column includes sequence-bootstrap 95% CIs, and the worst column is the largest harm rate among groups with at least 20 items. Values are percentages and `-Instruct` denotes an instruction-tuned checkpoint.

Task	Model	$\mathrm{acc}_{\mathrm{FP}}$	$\mathrm{acc}_{\mathrm{W4}}$	$\lvert\Delta\rvert$ (points)	flip (%)	harm (%) [95% CI]	worst (%)
MMLU	Llama-3.2-3B-Instruct	60.7	58.7	2.0	18.0	7.1 [6.7, 7.6]	13.0
	Llama-3.1-8B-Instruct	68.2	65.8	2.4	15.0	6.6 [5.7, 7.5]	19.2
	Qwen2.5-32B-Instruct	84.5	83.6	0.9	6.0	3.0 [2.4, 3.6]	9.6
	Qwen2.5-3B-Instruct	66.5	63.4	3.0	16.4	7.4 [7.0, 7.8]	16.0
	Qwen2.5-7B-Instruct	75.5	73.3	2.2	10.6	5.5 [4.7, 6.3]	15.4
HumanEval	Llama-3.2-3B-Instruct	59.1	52.4	6.7	21.3	14.0 [9.1, 19.5]	14.0
	Llama-3.1-8B-Instruct	67.7	62.2	5.5	16.5	11.0 [6.1, 15.9]	11.0
	Qwen2.5-32B-Instruct	87.8	87.8	0.0	12.2	6.1 [2.4, 9.8]	6.1
	Qwen2.5-3B-Instruct	75.0	50.6	24.4	35.4	29.9 [23.2, 37.2]	29.9
	Qwen2.5-7B-Instruct	80.5	78.0	2.4	17.1	9.8 [5.5, 14.6]	9.8
MBPP	Llama-3.2-3B-Instruct	63.4	56.8	6.6	16.7	11.7 [7.8, 15.6]	11.7
	Qwen2.5-3B-Instruct	66.9	55.6	11.3	24.5	17.9 [13.2, 22.6]	17.9
IFEval	Llama-3.2-3B-Instruct	65.3	65.7	0.4	18.6	9.1 [6.7, 11.7]	13.3
	Llama-3.1-8B-Instruct	71.5	69.5	2.0	16.2	9.1 [6.7, 11.7]	12.8
	Qwen2.5-3B-Instruct	63.0	58.0	5.0	20.8	12.9 [10.1, 15.8]	18.9
	Qwen2.5-7B-Instruct	73.1	69.3	3.8	16.8	10.3 [7.7, 13.1]	17.0

setting, the average also degrades sharply. Qwen2.5-3B-Instruct MMLU falls from 0.665 to 0.404, and code falls to near zero (Table 14).

Table 15: Harmful-flip rates under GPTQ-W3 for the same deployment benchmarks. All values are percentages.

Task	Model	harm_{W3} (%)
MMLU	Llama-3.2-3B-Instruct	26.4
	Llama-3.1-8B-Instruct	26.9
	Qwen2.5-32B-Instruct	10.3
	Qwen2.5-3B-Instruct	34.1
	Qwen2.5-7B-Instruct	19.0
HumanEval	Llama-3.2-3B-Instruct	56.1
	Llama-3.1-8B-Instruct	65.9
	Qwen2.5-32B-Instruct	33.5
	Qwen2.5-3B-Instruct	75.0
	Qwen2.5-7B-Instruct	73.8
MBPP	Llama-3.2-3B-Instruct	61.5
	Qwen2.5-3B-Instruct	66.9
IFEval	Llama-3.2-3B-Instruct	26.5
	Llama-3.1-8B-Instruct	27.3
	Qwen2.5-3B-Instruct	45.7
	Qwen2.5-7B-Instruct	26.1

J Extended Mixed-Precision Allocation Results

J.1 Allocation Rule and Shared Proxy Quantizer

Operational allocation rule. Modules have heterogeneous parameter counts, so every signal uses the same greedy, no-budget-overrun allocator; indivisibility can leave small, signal-dependent amounts of budget unused. Because the score is measured with one low-precision module at a time, the results in Sec. 4 and in this appendix characterize how the ranking performs when the deployed model contains many interacting low-precision sources.

Shared proxy quantizer. Every allocation signal, including impact, is computed once on a shared RTN proxy quantizer and then applied to whichever base is deployed; the deployed weights come from the genuine RTN, AWQ, GPTQ, or rotated base. All signals therefore see the same perturbation model, which isolates selection quality and keeps the signal cost base-independent.

J.2 Behavior Across Quantization Bases

Controls and bases. We compare butterfly-impact with five controls: RTN weight MSE; AWQ-style activation magnitude; the second-order Hessian signal used by HAWQ and GPTQ; the local reconstruction error $\mathrm{tr}(\Delta W H \Delta W^\top)$ that GPTQ and AWQ minimize; and the direction-averaged gain-weighted local proxy. Each comparison runs on five bases, ordered by local-error minimization strength: RTN, AWQ-style, GPTQ, randomly rotated GPTQ, and learned-rotation GPTQ. Table 16 reports the four strongest controls; the gain-weighted local proxy is in Table 17, and a random-allocation control is in Figure 6.

Cost–reliability frontier. Figure 6 shows bit-width and module selection as orthogonal axes of a single cost–reliability frontier: Impact dominates the interior on the rotated GPTQ base, whereas on plain GPTQ the curves cross and Hessian overtakes butterfly-impact at 3.5 bits, so held-out tail validation selects between the two regimes. The two views do not conflict: Table 16 averages hard-subset KL over Qwen2.5-1.5B and Qwen2.5-3B, whereas the frontier plots the upper-5% per-token KL CVaR_5 on Qwen2.5-3B alone, and on plain GPTQ the two metrics rank the top signals differently.

Table 16: Hard-subset KL across base quantizers at a matched parameter-weighted average bit-width budget of 3.5 bits, averaged over Qwen2.5-1.5B and Qwen2.5-3B. Every signal promotes modules from 3 to 4 bits under the same greedy, no-budget-overrun allocator, so only the score ordering differs. The hard subset is the top decile of per-sequence KL under the uniform base quantizer, fixed before any allocation rule is compared; entries are the hard subset's mean per-token $D_{\mathrm{KL}}(p_{\mathrm{FP}} \| p_{\mathrm{Q}})$ in nats, for which lower is better. Bases run left to right in order of how strongly each base already minimizes local error. Bold marks the lowest value in each column; Δ is butterfly-impact's relative improvement over the best competing signal, so negative values indicate worse performance. Dashes mark cells not measured.

	hard-subset KL, lower is better				
Allocation	RTN	AWQ-style	GPTQ	rot+GPTQ	learn-rot+GPTQ
act-mag	0.954	0.476	0.666	0.463	–
weight-MSE	1.296	0.504	0.469	0.427	–
Hessian	0.847	**0.436**	0.427	0.421	0.526
local-recon	**0.835**	0.440	0.435	0.426	0.521
butterfly-impact (ours)	0.904	0.444	**0.392**	**0.348**	**0.426**
Δ over best competing signal (%)	−8	−2	+8	+17	+18

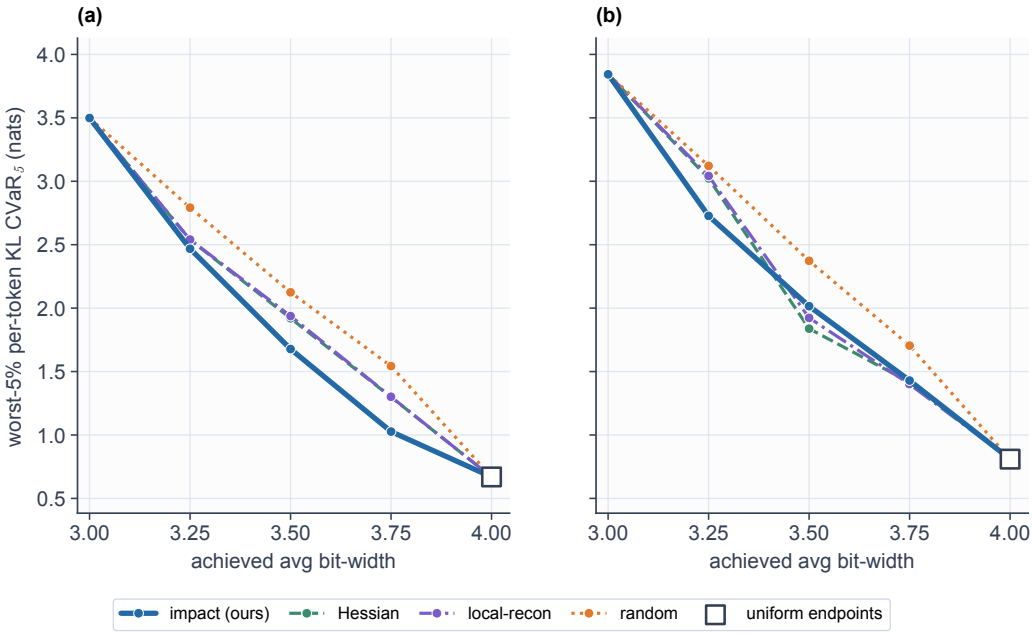


Figure 6: Bit-width and selection are orthogonal axes of one cost–reliability frontier. Qwen2.5-3B upper-5% per-token KL CVaR$_5$ against achieved average bit-width. Every signal is pinned at both ends, uniform 3-bit on the left and uniform 4-bit on the right, and differs only in the interior, where each signal chooses which modules receive the extra bit. Moving right lowers the curve at the cost of memory; moving down at a fixed x lowers CVaR$_5$ by reallocation, at the cost of the calibration sweep. The rotated GPTQ base appears in (a) and the plain GPTQ base in (b).

Bases with large local error. On the RTN base, local error is large and heterogeneous, so the local and second-order signals, which measure exactly that error, already pick good modules; butterfly-impact is competitive but not best, and the gap between signals is small.

Bases with low local error. On the GPTQ base, the quantizer has already driven local reconstruction error down across modules. The local and second-order signals saturate and become nearly indistinguishable, while the remaining errors differ in downstream propagation. There butterfly-impact cuts the hard-subset KL to 0.39, 8% below the best local and second-order Hessian baseline,

Table 17: Propagation-aware, tail-targeted mixed precision at a fixed parameter-weighted average bit-width target. The allocation promotes modules from 3 to 4 bits to reach an average of 3.5 bits; results are averaged over Qwen2.5-1.5B and Qwen2.5-3B. Every signal uses the same greedy, no-budget-overrun allocator, so only the score ordering differs. The table reports p99 and hard-subset per-input KL values, for which lower is better. On the GPTQ base, where local reconstruction error is already minimized, butterfly-impact achieves the lowest p99 and hard-subset KL. On the RTN base, local error dominates, and local and second-order signals perform best. Bold marks the lowest value in each column.

	RTN base		GPTQ base	
Allocation	KL_{p99}	$\mathrm{KL}_{\mathrm{hard}}$	KL_{p99}	$\mathrm{KL}_{\mathrm{hard}}$
act-mag (AWQ)	1.241	0.954	1.284	0.666
weight-MSE (RTN)	1.660	1.296	0.652	0.469
HAWQ and GPTQ Hessian	**1.052**	0.847	0.584	0.427
local-recon	1.052	**0.835**	0.584	0.435
gain-weighted local proxy	1.062	0.874	0.618	0.452
butterfly-impact (ours)	1.110	0.904	**0.502**	**0.392**

which reaches 0.43, at the same average bit-width, and lowers the p99 tail by 14% while matching PPL at 1.5B and improving PPL at 3B and 7B. The result is the mixed-precision counterpart of the causal result in Sec. 3.2: Accounting for downstream propagation matters most exactly when local error has been minimized. The allocation acts as a reliability layer applied on top of an existing quantizer.

Cross-base comparison in detail. Varying the base quantizer turns the contrast between GPTQ and RTN above into a cross-base comparison (Table 16). The bases are ordered by local-error minimization strength: RTN $\ll$ AWQ-style scale search $<$ GPTQ $<$ rotated GPTQ $\approx$ learned-rotation GPTQ. Butterfly-impact loses by 8% on RTN, ties within 2% on AWQ-style, and then wins by 8% on GPTQ, by 17% on randomly rotated GPTQ, where the p99 tail also falls by 16%, and by 18% on a SpinQuant-style learned-rotation base. The trained orthogonal rotation drives local error down further yet leaves the same propagating remainder for impact to catch. The last two bases coincide within noise, so the curve plateaus rather than continuing to climb. The pattern is consistent with the proposed mechanism. Propagation-aware selection becomes more useful after the base quantizer has substantially reduced local error, when the remaining error is more strongly distinguished by downstream effect. The single-model frontier of Figure 6 shows the same two regimes on a per-token metric. Impact trails the best local and second-order control at the 3.5-bit point on plain GPTQ by 10% in CVaR_5 and improves on the best competing control by 13% on rotated GPTQ, while the random-allocation control stays above every informed signal at each interior budget. For a new base, selection can therefore be assessed on a held-out tail before deployment.

Rotation-class bases. Rotation methods (Ashkboos et al., 2024; Liu et al., 2025) reshape the quantization coordinate system to shrink local error. We apply a randomized Hadamard residual-basis transform after folding RMSNorm scales; at group size 128, the residual-basis transform is an exact reparameterization, as Appendix H verifies to bf16 tolerance. The allocation comparison on the resulting GPTQ base is where propagation-aware selection wins by the largest margin (Table 16). The rotation makes the local signals nearly degenerate: Once the outlier structure is rotated away, Hessian and local-recon rank modules nearly identically, while butterfly-impact still finds the modules whose error propagates.

J.3 Robustness, Family Dependence, and Scale

Robustness across calibration draws. The comparison is stable under calibration resampling: Across three independent GPTQ calibration draws with the evaluation set fixed, butterfly-impact has the lowest hard-subset KL on both models. The detailed replication statistics and paired sequence-bootstrap intervals are in Appendix M.

Table 18: Qwen2.5-7B replication of the GPTQ mixed-precision comparison. Butterfly-impact gives the lowest hard-subset KL and PPL, while Hessian gives the lowest p99. Bold marks the lowest value in each column among the mixed-precision rows.

Allocation	KL_{p99}	$\mathrm{KL}_{\mathrm{hard}}$	PPL
uniform 4-bit	0.274	0.205	7.109
act-mag (AWQ)	0.626	0.548	8.261
HAWQ and GPTQ Hessian	**0.596**	0.529	8.109
local-recon	0.601	0.529	8.119
butterfly-impact (ours)	0.610	**0.506**	8.019

Family dependence. The comparison is, however, family-dependent: On Llama-3.2-1B at the same budget, butterfly-impact reaches a hard-subset KL of 1.475, below local-recon at 1.686 but above Hessian at 1.352. The Llama result is consistent with the cross-base trend: The propagation signal's edge is largest where the base has most thoroughly minimized local error, namely the Qwen family, the rotated base, and 7B as described below; on the smaller Llama-3.2-1B model, the curvature signal stays competitive. The cross-base and family comparisons locate the largest gains on bases where local-error signals have already saturated.

Scale. At 7B, Table 18 reports Hessians that are layer-chunked in host memory. On the GPTQ base at the matched budget, butterfly-impact gives the lowest hard-subset KL, at 0.506, compared with 0.529 for Hessian and 0.529 for local-recon, and the lowest PPL, at 8.02, compared with 8.11 for Hessian. On the broader p99, butterfly-impact is within 2.3% of Hessian, at 0.610 and 0.596, respectively. The hard-subset advantage persists at 7B, while the p99 difference is not statistically meaningful. On the weak RTN base at 7B, local signals win, as at every scale.

K Beyond Weights: KV-Cache Quantization

Long-context serving also compresses the KV cache. The attention-routing mechanism of Appendix E makes a sharp, testable prediction: A key error enters before the softmax and is a routing perturbation, so a key error can change which token attention reads, gated by the margin between the top two scores, whereas a value error enters after routing and is a content perturbation that cannot flip routing at all. Key and value errors should therefore have different downstream tails at matched local error.

Setup. We simulate KV-cache quantization at the attention interface on the post-RoPE, pre-replication key and value states, the tensors a quantized cache stores, using the literature's granularity (Liu et al., 2024b; Hooper et al., 2024): asymmetric min-max quantization, per-channel keys with scales over 32-token blocks, and per-token values with channel groups of 128. Streaming leaves each trailing partial key block in FP, and a per-token key mode is retained for the matched-form asymmetry test; Appendix H gives the simulation details.

Phenomenon. KIVI-style 4-bit KV-cache quantization, which we call KV4, quantizes only the KV cache while the weights stay FP. We observe the same butterfly signature as in weight quantization despite a mean PPL cost of only 1.3%: an 8.5× per-token KL tail ratio, a top-1 flip rate of 6% to 7%, measurable attention routing flips with 4.5% of all query-head top-1 decisions changing, and bifurcation in 90% of greedy generations. The butterfly effect is a property of the inference dynamics, not of which tensor is rounded.

Routing and content. The asymmetry test holds the quantizer form fixed: per-token, asymmetric, and with identical groups. The matched-form test quantizes K-only or V-only at 4 bits. Although the two tensors carry comparable relative error, at 0.15 for K and 0.12 for V, our matched-form test reveals substantially different output effects. V-only changes PPL by $+0.8\%$ and has zero routing flips by construction. K-only raises PPL to nearly 4×10^4 despite comparable local relative error. K-only quantization flips 19% to 38% of top-1 routing decisions and changes 95% to 98% of teacher-forced next-token predictions.

Table 19: Quantization tail failure on real zero-shot tasks under GPTQ-W4, averaged over Qwen2.5-1.5B and Qwen2.5-3B. A prediction flip disagrees with the FP prediction; a harmful flip changes a correct FP prediction into an incorrect quantized-model prediction. Average accuracy moves by at most about six points, and on four of the six tasks by far less, yet a much larger share of individual decisions changes; the same pattern persists at 7B and 14B under RTN-W4 (Table 20).

Task	acc_{FP}	acc_{Q}	flip (%)	harm (%)
LAMBADA	0.639	0.575	12.6	9.5
HellaSwag	0.557	0.558	10.6	4.2
ARC-Challenge	0.457	0.396	16.9	8.7
ARC-Easy	0.703	0.684	14.5	6.3
PIQA	0.773	0.769	7.2	3.8
WinoGrande	0.668	0.640	17.2	10.0

Mechanism of key failures. In the KIVI-style KV4 condition, consistent with the margin mechanism, the routing decisions that flip sit at a median score margin about 15 times smaller than the population median: 0.10 compared with 1.48. Consistent with the anisotropic-noise analysis of post-RoPE keys, the large divergence comes from outlier key channels stretching the per-token quantization range so the induced score noise dwarfs typical margins.

Granularity and scale. Per-channel key scales, a common choice in KV-cache quantizers (Liu et al., 2024b; Hooper et al., 2024), reduce the routing–content asymmetry. Per-channel 4-bit keys are as benign as 4-bit values. The result provides a mechanistic explanation for per-channel key scaling. Per-channel key quantization works because the per-channel scheme shrinks the routing perturbation, not merely the tensor error. The pattern persists at scale. On Qwen2.5-7B, KV4 costs 0.9% PPL yet keeps a $9.0\times$ tail, 3.9% routing flips and 83% generation bifurcation, and the per-token key and value asymmetry is nearly six orders of magnitude: $+3.8 \times 10^5\%$ compared with $+0.4\%$ PPL; at 14B, the KV4 tail reaches $13.3\times$ behind a 2.1% average cost, providing another high-tail KV scale point. This KV sweep is not monotone in size.

Long contexts and lower bit-widths. The routing–content asymmetry also persists at context lengths for which KV memory becomes a bottleneck. On 4096-token evaluation windows, KV4 keeps an $8.1\times$ tail with 5.2% routing flips behind a 1.3% mean cost, and the asymmetry remains intact, with per-token key quantization flipping attention routing on 31.0% of queries, compared with 0.0% for values. At 2 bits, the ordering reverses. With each tensor's best granularity, per-channel 2-bit keys hurt less than per-token 2-bit values. Once the routing perturbation is controlled, the remaining failure is content-dominated. The same ordering guides the choice of key and value bit-widths under a fixed average bit-width budget: Over Qwen2.5-1.5B and Qwen2.5-3B at 3.0 bits, an allocation with 2-bit keys and 4-bit values costs $+21.3\%$ PPL, while an allocation with 4-bit keys and 2-bit values costs $+91.9\%$. Under this evaluated average bit-width budget of 3 bits, assigning the lower precision to the granularity-protected keys yields lower PPL than assigning the lower precision to values.

L Additional Task-Level Analyses

L.1 Per-Model Task-Flip Rates

Table 19 reports the task-level aggregates discussed in Sec. 3.3, and Table 20 expands the task-level aggregates: Each reported maximum corresponds to a cell here, alongside the full distribution across models and tasks.

L.2 GSM8K CoT Results and Qualitative Butterfly Cases

This section extends the task-level audit of Sec. 3.3 with GSM8K greedy CoT results and qualitative butterfly cases.

Table 20: Per-model prediction-flip and harmful-flip rates underlying Table 19. This first block reports LAMBADA, HellaSwag, and ARC-Challenge; all values are percentages.

Model	Quantizer	LAMBADA		HellaSwag		ARC-Challenge	
		flip	harm	flip	harm	flip	harm
Qwen2.5-1.5B	GPTQ	15	12	12	5	16	8
	RTN	13	8	14	7	15	6
Qwen2.5-14B	RTN	7	4	7	4	16	5
Qwen2.5-3B	GPTQ	10	7	9	3	18	10
	RTN	17	13	16	9	23	10
Qwen2.5-7B	RTN	10	5	7	4	12	6

Table 20: Per-model prediction-flip and harmful-flip rates (continued): ARC-Easy, PIQA, and WinoGrande.

Model	Quantizer	ARC-Easy		PIQA		WinoGrande	
		flip	harm	flip	harm	flip	harm
Qwen2.5-1.5B	GPTQ	15	6	9	4	22	13
	RTN	15	7	8	4	23	12
Qwen2.5-14B	RTN	9	2	3	2	16	8
Qwen2.5-3B	GPTQ	14	7	6	3	13	7
	RTN	17	8	9	6	20	10
Qwen2.5-7B	RTN	6	3	6	3	16	10

Generative reasoning compounds the flips. Multiple-choice scoring isolates single decisions; deployment failures compound through autoregressive generation, where Sec. 3.1 showed nearly every trajectory eventually bifurcates. To connect single decisions to full generations, we evaluate GSM8K (Cobbe et al., 2021) with Qwen2.5-3B-Instruct using greedy CoT decoding on 300 test problems and identical batching across conditions; Table 21 reports the results. Our GSM8K results show a matching and amplified version of the multiple-choice pattern: GPTQ-W4 moves average accuracy from 0.847 to 0.800, yet 22% of the final answers change relative to the FP model, which is the user-visible consequence of a CoT trajectory that bifurcated mid-derivation.

Mean accuracy masks harmful flips. At W4, harmful flips from correct FP answers to incorrect quantized-model answers occur for 9.7% of problems. Beneficial flips occur for 5.0% of problems and partly offset the harmful flips, allowing average accuracy to conceal the regressions; at W3, accuracy collapses to 0.030, with 81.7% harmful flips against 0.0% beneficial flips. Aggregate benchmark accuracy and agreement with the deployed FP model are different properties, and standard practice usually measures only aggregate accuracy. The token-level anatomy below follows one FP-correct to W4-wrong case from the shared prefix to the first visible decision change and connects the branch to the exact barrier of Theorem 2. Most shared steps are certified, while the branch occurs at a near-tied, uncertified step (Figure 7).

Token-level anatomy of a harmful flip. Figure 7 follows one FP-correct to W4-wrong case from the shared prefix to the first visible decision change. The figure connects the observed branch to the exact output-distribution barrier of Theorem 2. Most shared steps are certified, while the branch occurs at a near-tied, uncertified step. The first changed token marks where the generated paths become observably different; continued computation also uses different weights, so the later arithmetic error is not attributed to the punctuation choice alone. Among the nine harmful flips in a 128-problem screening subset, we select the case with the longest shared greedy prefix and re-verify the selected case at batch size one.

A strategy-changing token. The first problem states that Wendi feeds each of 20 chickens three cups per day, with 15 cups used in the morning and 25 cups in the afternoon, and asks for the amount

Table 21: GSM8K greedy CoT under quantization for Qwen2.5-3B-Instruct on 300 problems. GPTQ-W4 stays within 4.7 percentage points of FP accuracy while changing 22.3% of final answers. The flip column is the fraction of final answers that differ from the FP answer, the harm column is the fraction of problems whose correct FP answer becomes incorrect, with bootstrap 95% CIs, and the div column is the fraction of problems whose greedy continuation departs from the FP continuation within the 512-token decoding budget.

Weights	acc	flip (%)	harm (%)	div (%)
FP (bf16)	0.847	–	–	–
GPTQ-W4	0.800	22.3	9.7 [6.3,13.0]	100
RTN-W4	0.687	32.7	19.7 [15.3,24.3]	100
GPTQ-W3	0.030	96.7	81.7 [77.3,86.0]	100

used in the final meal. The gold answer is 20. Both models generate the identical first 38 tokens, ending mid-sentence with `1. Calculate the total amount of feed given`. The FP model continues with `in the morning and afternoon:`, computes $15 + 25 = 40$, the daily requirement $20 \times 3 = 60$, and answers $60 - 40 = 20$, which is correct. The W4 model continues with `per chicken per day:`, re-reads the problem as three cups per meal and calculates $3 \times 3 = 9$ per chicken, or 180 in total, then answers 140. The two trajectories differ by one next-token decision at one low-margin position; everything downstream, including which facts of the problem get used, follows from the routing change.

First-token divergence. The second problem states that James runs three 60-meter sprints three times a week; the gold answer is 540 meters. At the first generated token, the FP trajectory uses `To determine`, whereas W4 uses `To find`. The substitution is a pure paraphrase carrying no semantic content, yet the W4 trajectory later inserts the spurious premise `3 sets of 3 sprints each day` and answers 3780, 7× the gold answer. Bifurcation is not confined to hard problems: Once the prefix differs at all, the conditioned computation differs.

M Replication and Calibration-Draw Robustness

Each replication redraws the calibration windows while holding the reported evaluation set fixed. Replication summaries are indexed by the number of independent draws rather than individual seed identifiers.

Phenomenon and Fisher replication. On Qwen2.5-3B GPTQ-W4, across 7 independent calibration draws, the tail ratio is 9.0× with SD 0.09. The top-1 flip rate is 14.5% with SD 0.20 percentage points, and the mean PPL gap is 6.8% with SD 0.37. On Mistral-7B, the corresponding 3-draw check gives a tail ratio of 9.1× with SD 0.15, a flip rate of 8.7% with SD 0.13 percentage points, and a mean PPL gap of 4.2% with SD 0.45. For the Fisher comparison, the 2-draw check gives Spearman 0.95 ± 0.05 and p99 predicted-to-actual ratio 0.94 ± 0.11.

Mitigation replication. Across three independent GPTQ calibration draws, butterfly-impact has the lowest hard-subset KL on both Qwen models. Table 22 reports the across-draw summary and the paired sequence-bootstrap CI; the paired CI is computed within a matched calibration draw, so the within-draw sequence-level point estimate need not equal the difference of the across-draw means.

N Input-Adaptive Precision

A learned propagation-aware probe. We train a light probe using logistic regression (LR) and a gradient boosting machine (GBM), as reported in Table 23, on a small vector of propagation-aware FP features, each motivated by one factor of the decomposition: per-layer downstream-gain-weighted residual energy, margins between the top two logits, output entropy, and an activation outlier ratio. To match the realistic offline-profiling setting, the probe is trained on a profiled C4 stream and tested on disjoint sequences from the same distribution, and the probe target is the same top-decile butterfly

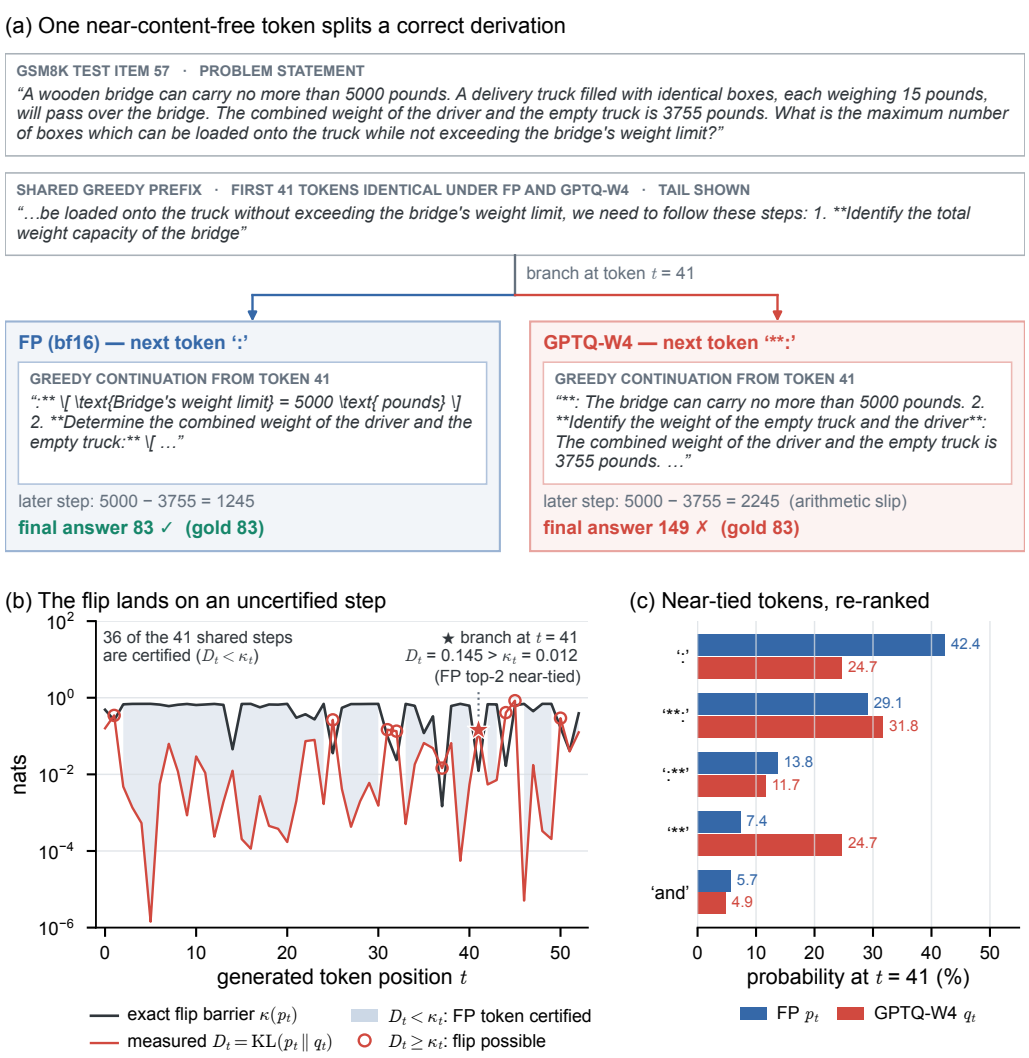


Figure 7: Anatomy of a CoT bifurcation at an exact low-barrier step. (a) On one GSM8K problem, Qwen2.5-3B-Instruct in bf16 and under GPTQ-W4 produce the same greedy prefix for 41 tokens, then diverge at a near-tied formatting token, : under FP and ∗∗: under W4. FP goes on to the gold answer 83, whereas W4 later miscomputes $5000 - 3755$ as 2245 and answers 149. (b) At the branch, the measured KL of 0.145 exceeds the exact barrier of 0.012 from Theorem 2; shading marks the 36 of 41 shared steps certified by $D_t < \kappa_t$, and the star marks the branch. (c) Next-token probabilities at the branch under FP and W4. The later arithmetic error reflects the full quantized continuation, not the branch token alone.

input under GPTQ-W4. We find that the learned probe reaches an area under the receiver operating characteristic curve (AUROC) of 0.86 [0.81,0.91] on Qwen2.5-3B using 384 held-out inputs and reaches 0.83 to 0.89 from 1.5B to 7B. The strongest single features are the propagation quantities. Gain-weighted residual energy reaches AUROC 0.76.

Risk routing. The learned probe enables a direct intervention: Routing the probe’s predicted-risky top decile to a higher precision removes 26% of the upper-5% per-token serving tail, against only 3% for a matched-budget random control, yielding about a ninefold larger tail reduction at the same average cost. The probe therefore supports input-level precision allocation when used with propagation-aware features and a learned ranker rather than a single low-cost score.

Table 22: Calibration-draw robustness of the GPTQ allocation result in Table 17. Cells give the mean $\pm$ standard deviation (SD) over three calibration draws; the last row gives paired sequence-bootstrap 95% CIs for the Hessian−impact difference, where positive favors impact.

	hard-subset KL, lower is better	
Allocation	Qwen2.5-1.5B	Qwen2.5-3B
uniform 4-bit	0.206 ± 0.007	0.117 ± 0.003
Hessian	0.500 ± 0.017	0.344 ± 0.013
local-recon	0.507 ± 0.016	0.354 ± 0.010
butterfly-impact (ours)	0.473 ± 0.003	0.329 ± 0.017
Δ(Hessian−impact) hard	$+0.049$ $[+0.019, +0.078]$	$+0.021$ $[+0.008, +0.035]$

Table 23: Input-risk prediction and precision routing. The probe predicts the top decile of per-sequence GPTQ-W4 KL from FP-only features. Brackets are sequence-bootstrap 95% CIs; the last two columns compare equal-budget learned and random routing.

	butterfly-input AUROC			upper-5% tail cut (%)	
Model	single best	learned (LR)	learned (GBM)	learned	random
Qwen2.5-3B	0.76	0.86 [0.81, 0.91]	0.80 [0.73, 0.86]	26	3
Qwen2.5-1.5B	0.78	0.83 [0.76, 0.89]	0.76 [0.66, 0.84]	28	4
Qwen2.5-7B	0.81	0.89 [0.85, 0.93]	0.87 [0.82, 0.92]	22	3
Qwen2.5-3B probe trained on C4 and tested zero-shot					
WikiText-2	0.74	0.82 [0.72, 0.90]	0.71 [0.61, 0.81]	8	2
code	0.84	0.75 [0.66, 0.84]	0.72 [0.58, 0.85]	23	3

Cross-distribution transfer. Frozen after training on C4 and applied zero-shot, the learned probe still ranks the GPTQ-W4 tail at AUROC 0.82 on WikiText-2 and 0.75 on code for Qwen2.5-3B. Each corpus is scored against a corpus-specific top decile, fixing the base rate at 10%. Routing the probe's predicted-risky decile on the unseen distribution removes 8% of the upper-5% tail on WikiText-2 and 23% on code, compared with 2% and 3%, respectively, for a matched random control (Table 23).

Scale dependence. The propagation-aware features thus carry risk information that transfers to distributions the probe never profiled, and the transfer strengthens with model scale: Repeating the same zero-shot transfer at 1.5B and 7B, which Table 23 does not tabulate, raises the WikiText-2 AUROC to 0.94 at 7B. The transfer is not uniform across corpora. On the tabulated Qwen2.5-3B cells the linear probe transfers cleanly to WikiText-2, at AUROC 0.82 against 0.74 for the best single feature, whereas on the more distant code distribution the linear probe's AUROC falls to 0.75 and the best single feature is the stronger ranker, at 0.84. Learning a combination therefore helps in-distribution and on a near corpus but not on the farthest corpus, which marks the residual limit of zero-shot cross-distribution transfer.

O Extended Discussion and Limitations

O.1 Deployment Guidance

Sec. 5 summarizes the two mitigation methods. This section gives practical deployment guidance and adds the input-adaptive third axis.

Our framework provides two complementary interventions, PAPA and BART, each with a practical deployment guideline. PAPA protects high-impact modules under a fixed bit budget. Use held-out tail validation to choose among butterfly-impact, local, and curvature allocation rules. Butterfly-impact leads in hard-subset KL on GPTQ and rotated GPTQ bases, whereas local controls lead on RTN and AWQ-style bases and remain competitive in per-token CVaR_5 on plain GPTQ (Figure 6 in Appendix J). At a fixed aggressive bit-width, BART is the training-time intervention to use when

degraded average performance or multi-step reasoning must be recovered. Our framework traces large output shifts and harmful prediction flips to the quantization errors amplified inside the model and provides targeted ways to reduce quantization-induced output errors and harmful flips.

Running the allocation rule. PAPA ranks modules by the measured single-source KL score, obtained by quantizing one module from FP to the low-bit setting on calibration data, and admits the ranked modules greedily under a parameter-weighted average bit-width budget. Because the ordering is base-dependent, the allocation rule must be validated empirically: Compare butterfly-impact against local and curvature controls on held-out tail data for the base actually being shipped. Qwen2.5 plain GPTQ is the cautionary cell, where Hessian stays token-competitive and butterfly-impact costs 10% in CVaR_5.

Input-adaptive precision. A third axis allocates precision per request rather than per module. A light propagation-aware probe reaches an AUROC of 0.86 and removes 26% of the upper tail when the probe's predicted-risky decile is routed to a higher precision, against 3% for a random control at the same budget (Table 23). Frozen after training, the probe transfers zero-shot across a distribution shift, with AUROC 0.82 on WikiText-2 and 0.75 on code, so the risk signal is not tied to the calibration corpus.

Scope of the training-time claim. Theorem 9 characterizes an effective additive rank-r correction under fixed Jacobian and Fisher geometry, which real training does not enjoy; Sec. 4.2 is the evaluation that puts the procedure through threshold crossings and input-dependent propagation. BART requires a short training pass and an FP teacher, and BART's largest gains occur at low bit-widths.

O.2 Limitations

Our experiments span several model families, quantization settings, and natural-language and code evaluation datasets, but the experiments do not cover every model, dataset, or deployment configuration. The reported empirical findings should therefore be interpreted within the evaluated scope; extending the same analysis to additional architectures, scales, data distributions, and workloads is a useful direction for future work.