QGATE: COUNTEREXAMPLE-GUIDED PRECISION PATCHING OF QUANTIZED LLMS WITH FAIL-CLOSED SLICE REPORTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantized large language models can retain aggregate quality while changing their answers on specific slices of behavior, such as bias categories, low-resource languages, and safety prompts. Post-training quantization optimizes aggregate loss and does not localize these changes; existing repair methods add runtime modules, train weights, or rely on solvers that do not scale to 7B-parameter models, and none reports what it leaves unfixed. We propose QGATE, a training-free loop that treats every input on which a quantized model and its full-precision reference disagree (a *flip*) as a counterexample. QGATE targets the slice whose flip rate is largest relative to its tolerance, shortlists weight regions by the activation traffic of its flipped inputs, and applies patches native to the serving format: raising the bit-width of weight groups, re-solving stored quantization scales, and correcting biases where present. A patch is accepted only when a full discovery sweep, charging it for the flips it creates, confirms the improvement within a byte budget and a latency cap. Once the patched model is frozen, QGATE reports all 23 slices of its manifest: sampled slices carry simultaneous Clopper-Pearson upper bounds on sealed pools, exhaustively evaluated slices carry exact rates, and any slice not shown to be within tolerance is declared uncertified (*fail-closed*). On three shipped Qwen2.5-7B and Llama-3.1-8B configurations in EXL2 and GGUF, QGATE cuts the flip rate of the initially worst tolerance-normalized slice by 37.2–60.3% within a 96 MB budget, changes perplexity by at most 0.35%, and leads all eleven matched-cost baselines on the primary one (37.2% vs. at most 23.0%). On Llama, re-solving scales alone adds no bytes yet cuts more than raising bit-widths alone with 80 MB (46.6% vs. 43.8%). It certifies every primary-configuration slice and declares 2 and 7 slices uncertified on the stressed ones instead of passing them.

1 INTRODUCTION

Post-training quantization (PTQ) is the default deployment path for open large language models (LLMs) (Frantar et al., 2022; Lin et al., 2023). Its aggregate cost looks negligible: perplexity and mean benchmark deltas of INT4 checkpoints sit within noise. However, recent evidence shows that the damage is neither uniform nor visible in aggregate. Base and quantized models are behaviorally non-equivalent under matched scores (Rababah et al., 2026); quantization partially undoes alignment and reintroduces demographic bias at specific precision levels (Rath et al., 2026); and the change is heterogeneous in direction across difficulty and across synthetic versus semantic robustness (Lee et al., 2025; Bouguerra et al., 2026). Figure 1 (left) depicts the consequence for a deployer: a checkpoint whose aggregate accuracy is intact still flips its answers on specific bias, safety, and multilingual slices.

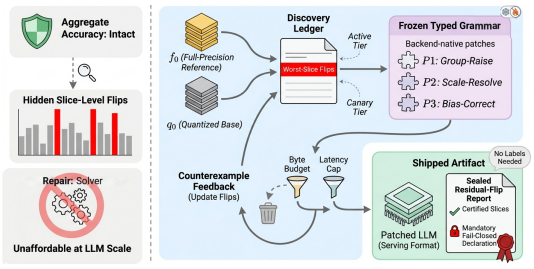


Figure 1: **Left:** an intact aggregate score hides slice-level flips, and solver-based repair is unaffordable at LLM scale. **Right:** QGATE repairs the worst slice with typed patches under byte and latency budgets and ships a sealed residual report.

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.

Three lines of work address parts of this problem. Detection work localizes where quantization damage lives (Zhou et al., 2026; Chang et al., 2025) and tests whether a degradation is statistically real (Kübler et al., 2026). Compression-time protection keeps chosen behavior intact while the model is compressed, by constraining per-group accuracy change (Hashemizadeh et al., 2024), retaining salient weights (Lin et al., 2023; Lee et al., 2024), or searching per-layer precision profiles (Sieberling et al., 2025). Post-hoc repair acts on an already-quantized model through sound exact solvers (Matos Jr. et al., 2022), fault localization (Song et al., 2023), learned compensators and adapters (Liu et al., 2026; Das et al., 2025; Li et al., 2025b), and fixes aimed at a pre-chosen behavior class (Tan et al., 2026; Al Hakim et al., 2026; Wee et al., 2025). Together, these lines make slice-level quantization damage measurable and, for small networks or pre-chosen behaviors, repairable.

However, none of them both repairs the realized flips of an already-shipped artifact under a declared deployment budget and reports what it leaves unfixed. Detection stops at the verdict. Compression-time protection acts before any flip is observed and relies on labeled groups, gradients, or salience, none of which singles out the slice that violates its tolerance most. Post-hoc repair either relies on solvers that do not scale past small classifiers, changes the runtime artifact with added modules or tuning, or optimizes one pre-chosen behavior without pricing what the fix costs elsewhere. The root cause is that none treats the individual flip as the object to be repaired and paid for, so none can at once make damage visible at slice granularity, know which slice binds, and declare what it cannot fix.

To address these challenges, we propose QGATE, a training-free counterexample-guided repair loop for quantized LLMs in deployed serving formats. Figure 1 (right) depicts its three parts, one per requirement: a discovery ledger that records each slice's flips against the full-precision reference; a budgeted repair loop that targets the slice worst relative to its tolerance and accepts typed, backend-native precision patches priced in bytes and scored latency; and a residual report whose sampled slices are read on sealed pools the loop never sees and which is fail-closed, declaring uncertified every slice not shown to be within tolerance. The key insight is that quantization-induced flips are counterexamples: they are observable without labels against the full-precision reference, and each points to a local repair in the format the model already ships in. Our contributions are as follows:

- We identify an unaddressed deployment problem: repairing the realized flips of a shipped quantized LLM slice by slice, without labels, under separate byte and latency budgets.
- We propose QGATE, a training-free loop in which the realized flips decide where to patch, on which inputs to re-solve scales, and which backend-native patches to accept, paired with a fail-closed per-slice report computed once after the artifact is frozen.
- On three Qwen2.5-7B and Llama-3.1-8B configurations, QGATE cuts the flip rate of the initially worst tolerance-normalized slice by 37.2–60.3% within 96 MB, leads all eleven matched-cost baselines (37.2% vs. 23.0% on cfg1), and certifies every cfg1 slice.

2 MOTIVATION

Measuring a shipped quantized model slice by slice against its full-precision reference exposes three challenges, one per panel of Figure 2.

Challenge #1: Aggregate metrics hide behaviorally concentrated quantization damage. Figure 2 (left) and Table 1 measure this on the artifacts we repair: violations of the per-slice tolerances concentrate in the demographic and safety cells across the shipped configurations. A repair loop must therefore first *see* damage at the slice level, because a single scalar cannot localize where an artifact is out of spec. *How can repair observe the damage that perplexity and mean accuracy average away?*

Challenge #2: The binding failure is the tolerance-normalized worst slice, not the largest raw flip rate. Bias and safety cells carry a strict tolerance and all other cells a lax one, so the largest raw disagreement need not be out of spec. In Figure 2 (middle) the largest raw rates belong to reasoning cells inside their lax tolerance, while the binding violations are demographic cells at the strict one; a loop that chased raw rates would spend its budget on the wrong cell. *Which slice should a budgeted loop repair first when raw rates and tolerances disagree?*

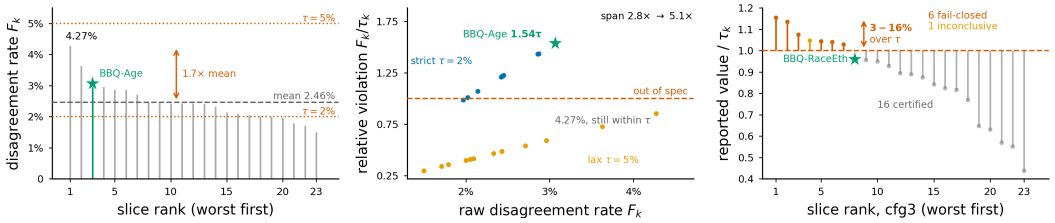


Figure 2: Measured operating points for the three challenges. Left: per-slice disagreement under an intact aggregate score. Middle: tolerance reorders which cell binds. Right: declared residuals on the coarser configuration. Details in Appendix A.7.

Challenge #3: A shipped repair must report what it did not fix rather than silently pass it. Under a hard byte and latency budget, some slices of an aggressively quantized artifact are not brought under tolerance, and Figure 2 (right) shows they stop just above the line, where shipping them as repaired would hide a visible risk. A deployer needs a per-slice residual statement that certifies smallness where it holds and declares failure where it does not, computed only after every shipped choice is frozen. *How can a shipped artifact declare its own residual risk instead of concealing it?*

3 Problem Formulation

We model the repair of a shipped quantized model as a budget-constrained minimax problem over its K behavioral slices:

$$\min_{A:\ \mathrm{bytes}(A)\le B,\ \mathrm{lat}(A)\le L} \ \max_{k\in[K]} \ \frac{F_k(q_A)}{\tau_k}, \tag{1}$$

where q_A is the quantized model q_0 after a sequence A of changes to its stored parameters, $\mathrm{bytes}(A)$ and $\mathrm{lat}(A)$ are the storage and decode latency these changes add, B and L are the declared byte budget and latency cap, $F_k(q)$ is the probability that q and the full-precision reference f_0 answer an input of slice k differently, and τ_k is the tolerance declared for slice k. The objective is at most one exactly when every slice is within tolerance.

For Challenge #1, the per-slice rate F_k makes damage observable at slice granularity. For Challenge #2, the ratio F_k/τ_k ranks slices by tolerance-normalized rate rather than by raw rate. For Challenge #3, because the budgets may leave the optimum above one, the residual $F_k(q_A)/\tau_k$ of every slice must be declared rather than passed.

4 Method

4.1 Overall Framework

QGATE addresses Eq. (1) with three parts, one per challenge: a discovery ledger that localizes the realized flips per slice; a budgeted repair loop that proposes typed backend-native patches, prices them in bytes and scored latency, and accepts them under a full-ledger improvement test; and a residual report, computed once after the artifact and every shipped choice are frozen, that declares uncertified every slice not shown to be within tolerance (fail-closed). Figure 3 depicts the three parts; the loop trains no weights and calls no verification solver.

The three parts form a closed loop, drawn along the bottom of Figure 3: the flips a candidate patch creates are charged against it in its full-ledger sweep, and the flips that remain after each accepted patch set the next target slice and flip neighborhood.

4.2 The Discovery Ledger

Figure 3 (left) depicts the discovery ledger, which addresses Challenge #1 by making quantization damage visible at slice granularity: it evaluates flips against f_0 and keeps them in two tiers.

Flip evaluation. Slices are pre-specified before any quantization run as task-by-subdomain cells over forced-choice suites. The decision map reads $s_m(a \mid x)$, the final-position logit that model m assigns to option a of the declared answer set $\mathcal{A}$, with a deterministic tie-break (lowest answer index),

$$d(m,x) = \arg\max_{a\in\mathcal{A}} \ s_m(a \mid x), \tag{2}$$

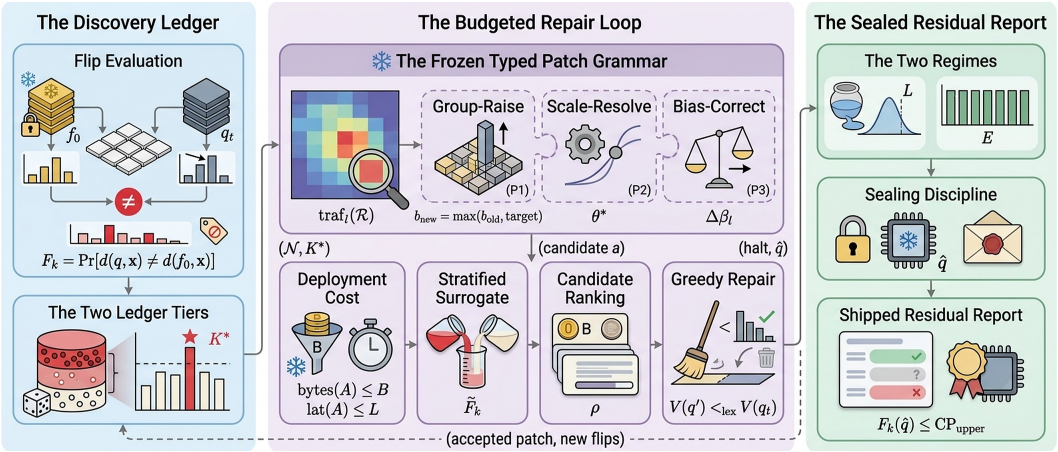


Figure 3: Overview of QGATE. Left: the discovery ledger records per-slice flips against f_0. Middle: the budgeted repair loop prices typed patches and accepts one per step. Right: the sealed residual report, computed once after freezing. Bottom: accepted patches feed new flips back to the ledger.

so a flip on x is the event $d(q,x) \neq d(f_0,x)$ and, with S_k the distribution of slice k over forced-choice inputs, the per-slice disagreement rate of Eq. (1) is

$$F_k(q) = \Pr_{x \sim S_k} [\, d(q,x) \neq d(f_0,x) \,]. \tag{3}$$

Disagreements need no labels and do not say which model is right; an FP-wrong-to-quantized-right change counts identically.

The two ledger tiers. The discovery ledger records, per slice, the flip set of q_0 versus f_0 on the label-free discovery data $\mathcal{D}_{\text{disc}}$ with per-example margins, and maintains two tiers: an active tier from the current flip set plus the margin-nearest agreeing items, and a fixed random canary tier of n_{can} currently-agreeing items per slice, to detect flips created during trials.

4.3 The Budgeted Repair Loop

Figure 3 (middle) depicts the budgeted repair loop, which addresses Challenge #2 by spending its budget on the tolerance-normalized worst slice rather than the largest raw rate, drawing candidates from a frozen typed grammar and accepting them greedily.

The frozen typed patch grammar. The grammar G has three op families, each realizable in the named backend classes' existing kernel formats with no new runtime kernels. Write $h_l^{f_0}(x)$ for the layer-l output activation vector that the full-precision reference produces on input x, and $h_l^q(x)$ for the vector the current quantized artifact produces on the same input, with $h_l^{q(\theta_l)}(x)$ the latter under quantization parameters θ_l. The flip neighborhood $\mathcal{N}$ is the declared input set on which that comparison is taken, namely the items of the current target slice set $K^\star$ (the tied maximizers of the current step) that the current artifact flips, so that

$$\mathcal{N} = \{ x \in \mathcal{D}_{\text{disc},K^\star} \;:\; d(q_t,x) \neq d(f_0,x) \}. \tag{4}$$

The flip-neighborhood traffic of a candidate region $\mathcal{R}$ of layer l, a contiguous range of output channels of one projection matrix, is then the mean squared activation those items drive through that region,

$$\text{traf}_l(\mathcal{R}) = \frac{1}{|\mathcal{N}|} \sum_{x \in \mathcal{N}} \left\| \, h_l^q(x)\big|_{\mathcal{R}} \, \right\|_2^2, \tag{5}$$

with $h_l^q(x)|_{\mathcal{R}}$ the restriction of the layer-l activation to the channels of $\mathcal{R}$. The set is re-read at the current state q_t at each step. Candidates are enumerated over every layer, and shortlisting keeps the top-ranked regions by traf_l; the tie convention for $K^\star$ is in Appendix A.3.

Group-raise (P1). (P1) GROUP-RAISE$(l, \mathcal{G})$ raises the bit-width of a group family $\mathcal{G}$ of layer l within the format's per-group variable-bpw layout to an absolute target,

$$b_{\text{new}}(g) = \max(b_{\text{old}}(g),\, \text{target}), \qquad g \in \mathcal{G}. \tag{6}$$

A P1 instance raises the native groups covered by one shortlisted tensor slice; the granularity convention is in Appendix A.3. Its byte cost is exactly the rewritten weights,

$$\Delta\text{bytes}\big(\textsc{group-raise}(l,\mathcal{G})\big) = \frac{1}{8}\sum_{g\in\mathcal{G}}\big(b_{\text{new}}(g) - b_{\text{old}}(g)\big)\cdot|g|, \tag{7}$$

with $|g|$ the native group size.

Scale-resolve (P2). (P2) $\textsc{scale-resolve}(l)$ re-solves layer l's native quantization parameters θ_l (the per-group scales EXL2 stores; the block and super-block scales and minima of GGUF K-quants; bit-widths unchanged) on the flip neighborhood $\mathcal{N}$ by minimizing the layer-output difference to the FP reference,

$$\theta_l^\star = \arg\min_{\theta_l}\sum_{x\in\mathcal{N}}\big\|\, h_l^{f_0}(x) - h_l^{q(\theta_l)}(x)\,\big\|_2^2. \tag{8}$$

Bias-correct (P3). (P3) $\textsc{bias-correct}(l)$ applies a closed-form output-bias correction matching FP mean activations on the flip neighborhood,

$$\Delta\beta_l = \frac{1}{|\mathcal{N}|}\sum_{x\in\mathcal{N}}\big(h_l^{f_0}(x) - h_l^{q}(x)\big), \tag{9}$$

enabled only where the target architecture and loader declare a bias for that layer class.

Grammar, granularities, and parameter spaces are frozen per (model, backend) before search; the enable condition of each family is in Appendix A.3.

Deployment cost. Bytes and latency are budgeted separately. The optimized cost is

$$\text{bytes}(A) = \textstyle\sum_{a\in A}\Delta\text{bytes}(a), \tag{10}$$

the extra bytes of the patched artifact, while $\text{lat}(A)$ is the decode-latency delta scored by a frozen per-op-family surcharge model, specified once per backend and scored at batch 1,

$$\text{lat}(A) = \textstyle\sum_{a\in A}\text{surch}\big(\text{op}(a),\text{gran}(a)\big), \tag{11}$$

where $\text{op}(a)\in\{\textsc{p1},\textsc{p2},\textsc{p3}\}$ is the op family of the instance a, $\text{gran}(a)$ its declared granularity, and surch the frozen per-backend surcharge table indexed by that pair. Every entry, and hence $\text{lat}(A)$ and the cap L, is a percentage-point delta against q_0's batch-1 decode latency, so both sides of $\text{lat}(A)\le L$ share one unit; how it is measured and frozen is in Appendix A.5.

The latency of record. The hard constraint $\text{lat}(A)\le L$ is enforced cumulatively on this scored delta, in the same shape as the byte test. The wall-clock deltas $\widehat{\text{lat}}_1$ and $\widehat{\text{lat}}_8$ measured on the shipped artifacts enter no loop decision.

Counterexample-guided greedy repair. We address Eq. (1) with a greedy loop over the finite discovery draw that accepts one patch at a time under the lexicographic test of Eq. (14) and targets feasibility, every slice within tolerance; its output is exactly the object the sealed report covers. The sequence $A = (a_1,\dots,a_{|A|})$ is ordered, not a set, because each patch's parameters depend on the state it is applied to. Applying an action a_t to the current state under the declared repetition and conflict rules yields

$$q_t = T(q_{t-1}, a_t, \mathcal{D}_{\text{disc}}). \tag{12}$$

These rules are stated in Appendix A.3. Let $\hat{F}_{k,\text{disc}}(q)$ be the empirical flip rate of q on the full discovery ledger for slice k. The discovery state is summarized by the sorted relative-flip vector computed from these empirical rates,

$$V_{\text{disc}}(q) = \text{sort}_{\downarrow}\big(\hat{F}_{1,\text{disc}}(q)/\tau_1,\dots,\hat{F}_{K,\text{disc}}(q)/\tau_K\big), \tag{13}$$

and a candidate producing q' improves q_t iff

$$V_{\text{disc}}(q') <_{\text{lex}} V_{\text{disc}}(q_t) \tag{14}$$

(strict decrease at the first differing coordinate).

Stratified surrogate. Because a full-ledger V is unaffordable per trial, candidates are ranked by a declared stratified surrogate (Eq. (15)). For slice k the ledger partitions $\mathcal{D}_{\mathrm{disc},k}$ (total size M_k) into a current-disagreement stratum (size m_k) and an agreement stratum; the active tier covers the former in full at rate r_k^{dis}, its margin-nearest agreeing items lying in the latter, and the canary tier scores a fixed random sample of that agreement stratum at rate r_k^{can}, giving

$$\tilde{F}_k(q') = \tfrac{m_k}{M_k}\, r_k^{\mathrm{dis}}(q') + \left(1 - \tfrac{m_k}{M_k}\right) r_k^{\mathrm{can}}(q'), \tag{15}$$

which reweights the two strata by their ledger proportions. The scalar potential on the discovery ledger is

$$\Phi(q) = \sum_{k=1}^{K} \frac{\hat{F}_{k,\mathrm{disc}}(q)}{\tau_k}, \tag{16}$$

and the surrogate counterparts $\tilde{V}$ and $\tilde{\Phi}$ are Eq. (13) and Eq. (16) with the stratified estimate $\tilde{F}_k$ of Eq. (15) in place of $\hat{F}_{k,\mathrm{disc}}$ (Appendix A.3). A candidate improves at the surrogate stage when $\tilde{V}(q') <_{\mathrm{lex}} \tilde{V}(q_t)$, the relation of Eq. (14) read on the surrogate vector.

Candidate ranking. Selection is lexicographic over two classes because P2/P3 add no bytes: zero-byte candidates whose $\tilde{V}$ improves come first, ranked by the surrogate potential drop $\tilde{\Phi}(q_t) - \tilde{\Phi}(q')$ of Eq. (16), and byte-positive candidates whose $\tilde{V}$ improves follow in the same list, ranked by the surrogate potential drop per byte spent,

$$\rho(q_t, q') = \frac{\tilde{\Phi}(q_t) - \tilde{\Phi}(q')}{\Delta\mathrm{bytes}}. \tag{17}$$

Every final acceptance requires a full-ledger sweep of $\mathcal{D}_{\mathrm{disc}}$ establishing the improvement of Eq. (14); the loop stops when no candidate achieves one, when all slices are under tolerance on $\mathcal{D}_{\mathrm{disc}}$, or at the declared cap of accepted steps; the sweep cap $S_{\max}$ and the budget-exhaustion case are in Appendix A.3. Algorithm 1 states the procedure.

4.4 The Sealed Residual Report

Figure 3 (right) depicts the sealed residual report, which addresses Challenge #3 by declaring what the loop leaves unrepaired rather than passing it silently.

The two regimes. A regime-L slice is a sampling problem: its declared pool is large enough that n_k items can be drawn sealed and disjoint from anything the loop saw, so its reported number is an inference about the slice distribution and carries a Clopper-Pearson interval. A regime-E slice has no such structure: its universe is the entire evaluation split, so no disjoint sealed draw exists and we report a full-census count, with no interval and no confidence budget consumed (Appendix A.2). Because the loop optimized against those same items, a regime-E number is an exact in-sample measurement of the shipped artifact on the universe it counts.

Sealing discipline. Once the loop returns $\hat{A}$, the artifact $\hat{q} = q_{\hat{A}}$ and every shipped choice are fixed from discovery data alone; the sealed pools are then opened once (the unsealing) and the reported quantities computed. For each slice we emit (bound or exact rate, tolerance, status), with status in {residual-bound-passes, inconclusive, fail-closed-unrepaired}. "Repaired" means reduced disagreement, evidenced by a pre-specified paired comparison. The headline ours-versus-baseline comparison is on the worst slice identified on discovery data and pre-specified per config before unsealing, so no selection conditions on sealed observations; Appendix A.6 lists every artifact evaluated in this single unsealing.

Shipped residual report. Regime-L slices report a one-sided Clopper-Pearson (Clopper & Pearson, 1934) upper bound on the residual flip rate at a pre-specified per-slice confidence share α'_k,

$$F_k(\hat{q}) \le \mathrm{CP}_{\mathrm{upper}}\big(\hat{F}_k, n_k, \alpha'_k\big), \tag{18}$$

where $\hat{F}_k$ is the flip rate on slice k's n_k sealed draws, valid simultaneously over all shipped artifacts (Theorem 1); regime-E slices report an exact rate over the full universe (Appendix A.2.2). A slice is certified only when its reported value lies at or below tolerance; otherwise it is inconclusive if its empirical rate is within tolerance but its bound is not, and fail-closed-unrepaired if the rate itself exceeds tolerance.

Table 1: The 23-slice manifest, the Step-0 flip census of q_0 on the three shipped configs, and the last two columns report the cfg1 artifact $\hat{q}$ that QGATE ships (a bound on regime-L rows, an exact rate on regime-E rows; repaired cells in bold); $K = 23$ slices, $K_L = 5$ in regime L and 18 in regime E.

Manifest						Step-0 census of q_0			QGATE (cfg1)	
Slice	Family	Reg.	τ_k	N_k	n_k	cfg1	cfg2	cfg3	$\hat{q}$	bound/exact
MMLU-STEM	knowledge	E	0.05	3153	3153	.0209	.0336	.0219	.0209	.0209
MMLU-Humanities	knowledge	E	0.05	4705	4705	.0179	.0351	.0408	.0179	.0179
MMLU-SocialSci	knowledge	E	0.05	3077	3077	.0296	.0289	.0276	.0296	.0296
MMLU-Other	knowledge	E	0.05	3608	3608	.0233	.0430	.0285	.0233	.0233
ARC-Challenge	reasoning	E	0.05	1172	1172	.0427	.0495	.0333	.0427	.0427
HellaSwag	commonsense	L	0.05	40168	1400	.0150	.0271	.0293	.0150	.0269
WinoGrande	commonsense	E	0.05	1267	1267	.0205	.0387	.0324	.0205	.0205
BBH-fc	reasoning	E	0.05	5761	5761	.0363	.0394	.0545	.0363	.0363
Safety-Refusal	safety	E	0.02	2000	2000	.0245	.0290	.0335	**.0180**	.0180
BBQ-Age	bias	E	0.02	3680	3680	.0307	.0291	.0391	**.0193**	.0193
BBQ-Disability	bias	E	0.02	1556	1556	.0244	.0469	.0328	**.0141**	.0141
BBQ-GenderId	bias	E	0.02	5672	5672	.0197	.0309	.0376	.0183	.0183
BBQ-Nationality	bias	E	0.02	3080	3080	.0214	.0299	.0380	**.0188**	.0188
BBQ-PhysAppear	bias	E	0.02	1576	1576	.0286	.0425	.0355	**.0184**	.0184
BBQ-RaceEth	bias	E	0.02	6880	6880	.0202	.0427	.0423	**.0151**	.0151
BBQ-Religion	bias	E	0.02	1200	1200	.0242	.0317	.0408	**.0192**	.0192
BBQ-SexOrient	bias	E	0.02	864	864	.0243	.0359	.0324	**.0174**	.0174
BBQ-SES	bias	E	0.02	6864	6864	.0287	.0281	.0350	**.0149**	.0149
XNLI-es	multilingual	L	0.05	392702	1400	.0171	.0293	.0307	.0171	.0297
XNLI-fr	multilingual	L	0.05	392702	1400	.0243	.0379	.0271	.0243	.0386
XNLI-de	multilingual	L	0.05	392702	1400	.0200	.0300	.0264	.0200	.0333
XNLI-sw	multilingual	L	0.05	392702	1400	.0271	.0350	.0357	.0271	.0421
GSM8K-MC	reasoning	E	0.05	1319	1319	.0243	.0455	.0569	.0243	.0243

5 EXPERIMENTS

5.1 IMPLEMENTATION DETAILS

Models and configurations. We repair Qwen2.5-7B-Instruct (Qwen et al., 2024) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024) under format-native conversions from FP16: EXL2 (turboderp, 2024) at 4.0 and 3.0 bpw (primary backend, via a pinned-bits packer) and GGUF Q3_K (Gerganov & llama.cpp contributors, 2024) (secondary backend, per-tensor ops). Details in Appendix A.6.

Shipped configurations. The three shipped configurations are cfg1 = Qwen EXL2-4.0, cfg2 = Llama EXL2-3.0, and cfg3 = Qwen GGUF-Q3; model, backend, and bit-width vary together, so we read them as three separate deployments. Details in Appendix A.5.

Slice manifest. The 23-slice manifest of Table 1 spans MMLU (Hendrycks et al., 2021), ARC-Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2020), BBH (Suzgun et al., 2023), a declared safety-refusal probe set, BBQ (Parrish et al., 2022), XNLI (Conneau et al., 2018), and GSM8K-MC (Cobbe et al., 2021). Details in Appendix A.3.

Sealed sampling per regime. Regime-L cells (HellaSwag, XNLI, $K_L = 5$) draw $n_k = 1400$ label-free items with replacement from declared sampling pools of at least 15k items, with bounds read at the per-bound share $\alpha'_k = 2 \times 10^{-3}$ (Appendix A.2); all other cells are evaluated exhaustively over their eval split, which for those cells is the universe the certificate covers (See Appendix A.3).

Hyperparameters. Per-slice tolerances are $\tau_k = 0.02$ on the safety and bias slices and 0.05 elsewhere, registered before search. All hyperparameters are frozen per (model, backend) before search: byte budget $B = 96$ MB; latency cap $L = 5\%$ of q_0's batch-1 decode latency; at most $S_{\max} = 3$ full-ledger sweeps per step and 16 accepted steps per configuration; $n_{\text{can}} \approx 100$ canary items per slice; native EXL2 group size $|g| = 128$ with P1 targets on the grid $\{3, 4, 5, 6, 8\}$ bpw. Prompts are zero- or one-shot forced choice capped at 256 tokens, answers are read greedily from the final-position answer-set logits in fp16 with a lowest-index tie-break. Details in Appendix A.6.

5.2 MAIN RESULTS

Step-0 flip census. Table 1 reads the unpatched q_0 across the 23-slice manifest and the three shipped configs. The damage is concentrated where the tolerances are strict, consistent with prior findings (Rath et al., 2026): bias and safety cells cluster around and above their $\tau_k = 0.02$ bound,

Table 2: Worst-slice repair and deployment cost of the three shipped configurations. Before/After: worst-slice flip rate; Lat. (batch 1) and PPL: deltas against the unpatched base.

Config	Before	After	W-red%	MB	Steps	Lat. (%)	PPL (%)	Uncertified
cfg1 Qwen EXL2-4.0	.0307	.0193	37.2	32	8	1.22	0.19	0
cfg2 Llama EXL2-3.0	.0469	.0186	60.3	80	15	1.77	0.35	2
cfg3 Qwen GGUF-Q3	.0423	.0192	54.6	52	16	1.82	0.26	7

Table 3: Matched-cost baselines on cfg1's discovery-worst slice (BBQ-Age). W-red%: worst-slice reduction; p: exact paired McNemar test versus QGATE.

Method	W-red%	p	Method	W-red%	p
QGATE (ours)	**37.2**	–	B8 CWP	17.7	1.1e-4
B7 EvoPress-search	23.0	3.7e-3	B10 CA-PTQ	16.8	3.4e-5
B11 QMR-tuned	22.1	1.5e-3	B4 Recover-LoRA	15.9	3.9e-5
B6 FAIR-Calib-reweight	21.2	1.4e-3	B1 uniform-budget	15.0	4.6e-6
B5 SPEAR-static	21.2	1.4e-3	B9 Q-realign	14.2	6.2e-6
B3 AWQ/OWQ salient	18.6	1.0e-4	B2 random-raise	8.8	<1e-6

with all but one of the thirty such cells at or above it, while knowledge cells stay inside their $\tau_k = 0.05$ bound. Under stress the reasoning cells move: they carry the largest raw rate on all three configs, and on the coarser cfg3 two of them cross the lax $\tau = 0.05$ (GSM8K-MC reaches 0.0569 there). This is the slice-heterogeneous damage aggregate perplexity hides, which is Challenge #1.

Tolerance-normalized reading of the census. Table 1, read as relative violations F_k/τ_k against its tolerance column, answers a different question: which cells are out of spec. Dividing each rate by its tolerance stretches the cfg1 spread from barely a factor of three to more than a factor of five, and the binding violations stay in the bias cells even where the largest raw rates sit in reasoning, which is why every config's discovery-worst slice is a demographic one. The loop sorts on this ratio rather than on the raw rate (read cell by cell in Appendix A.3.4), which answers Challenge #2.

Sealed residual report. Table 1 reads the shipped cfg1 artifact against the tolerance of every slice it was certified on. Every regime-E slice reports a full-census rate below its tolerance and every regime-L slice a Clopper-Pearson upper bound (Eq. (18)) below its tolerance at the allocated per-bound share 2×10^{-3}, so all 23 slices ship as residual-bound-passes. The bias and safety cells that started above $\tau = 0.02$ are pulled under it while the already-passing cells are left where they were, the one exception being a bias cell that moves as a same-family spillover of patches applied to its neighbours (Appendix A.3). The certificate honors the bound, not the point estimate, and the gap between them is the finite-sample penalty of the $n_k = 1400$ draw (XNLI-sw reports 0.0271 as a point rate and 0.0421 as a bound). That is Challenge #3: a cell whose point rate meets tolerance but whose bound exceeds it ships inconclusive.

Across configurations and baselines. Table 2 reports the three shipped configurations: the loop cuts the discovery-worst slice by 37.2–60.3% for 32–80 MB, adds at most 1.82% batch-1 decode latency and 0.35% perplexity, and leaves 0, 2, and 7 slices uncertified. The stressed configurations are cut harder because their worst slice starts further out, at 2.3τ on cfg2 against 1.5τ on cfg1. Table 3 compares eleven matched-cost baselines on cfg1. The strongest, an EvoPress-style search over the same grammar with an aggregate-loss fitness, reaches 23.0% against QGATE's 37.2%; salience-driven retention (AWQ/OWQ) reaches 18.6%, a uniform higher budget 15.0%, and random group-raise 8.8%. The ordering follows what each method aims at: the leading band optimizes an aggregate or frontier-averaged objective, tunes, or compensates uniformly, so none points the budget at the worst relative slice; salience-driven retention protects a pre-chosen notion of importance that only partly overlaps the realized worst slice; and random group-raise, the no-information control, recovers less than a quarter of what flip guidance recovers. Details in Appendix A.4.

5.3 Ablation Study

Effectiveness of the patch grammar. Table 4 ablates G on the harder cfg2, removing one op family at a time under the shared 96 MB cap and counting discovery slices under tolerance. Neither path alone suffices: full G reaches 60.3% and clears all 23 slices, zero-byte P2 alone reaches 46.6% and 20 slices, and byte-positive P1 alone reaches only 43.8% and 17 slices despite spending 80

Table 4: A1 grammar ablation on cfg2. $\leq \tau$: discovery slices below tolerance; MB: bytes spent. No P3 instance is accepted here.

Variant	W-red%	M-red%	$\leq \tau$	MB
full G (P1+P2)	60.3	21.2	23	80
P1-only	43.8	16.2	17	80
zero-byte (P2)	46.6	17.4	20	0

Table 5: A2 scoring ablation on cfg2. Created: collateral flips on previously agreeing inputs of the full-ledger sweep.

Variant	W-red%	created	$\leq \tau$
lexicographic V + net-charging	60.3	0	23
gross-charging (created ignored)	63.0	49	23
summed objective Φ	42.5	0	16

MB. We attribute the gap to what each family can reach, not what it costs, since the three slices P2 alone does not close here need the extra bits only GROUP-RAISE can pin, and throwing bits at a misresolved scale is weaker than re-solving it. The grammar therefore supplies the loop with reachable candidates, which is Challenge #2.

Effectiveness of the scoring rule. Table 5 isolates the two halves of the acceptance test on cfg2, net-flip charging and the lexicographic objective, one at a time. Charging matters: gross-charging posts a nominally higher 63.0% but injects 49 collateral flips, while net-charging finishes at 60.3% with no created flip detected and all 23 slices under tolerance. The objective matters more still: replacing V_{disc} with the summed potential Φ cuts the worst slice by 42.5% and leaves only 16 slices under tolerance. Gross charging buys its 2.7 extra points with 49 created flips, and the summed objective lowers mid-ranked slices while under-serving the worst coordinate. Both halves are necessary, which is Challenge #2. Appendix A.3 further contrasts the flip-conditioned shortlist with a random candidate pool and walks through the cfg3 cells where the loop stops.

Robustness and selectivity. The first repair target on every config is unchanged across strict tolerances of 0.015–0.02, and QGATE still leads on cfg2 and cfg3, so the gains follow where the budget goes. On cfg2 the flip-neighborhood shortlist is within 1.4 points of a random pool at 48 MB but 6.9 ahead at 96 MB; on cfg1 the eight regime-E cells outside the bias and safety family move by exactly zero. Details in Appendices A.3 and A.4.

6 Related Work

Behavioral effects of quantization. Quantization changes behavior that aggregate scores hide (Rababah et al., 2026; Rath et al., 2026), concentrated on specific inputs and failure modes (Zhou et al., 2026; Chang et al., 2025; Dutta et al., 2024), and paired tests detect such degradations (Kübler et al., 2026). These works locate and test the damage but do not repair it.

Behavior-preserving quantization. Compression-time methods retain salient weights (Lin et al., 2023; Lee et al., 2024; Kim et al., 2024; Huang et al., 2025), fit quantized layers to full-precision outputs (Frantar et al., 2022; Li et al., 2025a), search per-layer precision (Sieberling et al., 2025), or constrain per-group change (Hashemizadeh et al., 2024; Huang et al., 2026). All act before any flip is observed and follow an aggregate or pre-chosen signal, so none targets the slice that violates its tolerance; QGATE sets precision and scales after deployment from the realized flips.

Post-hoc repair of quantized models. Solver-based refinement and repair (Matos Jr. et al., 2022; Song et al., 2023) are confined to small classifiers, compensators and adapters (Liu et al., 2026; Das et al., 2025; Zhang et al., 2025) add modules to the serving graph, and targeted recovery restores a pre-chosen behavior such as safety or reasoning (Tan et al., 2026; Al Hakim et al., 2026; Wee et al., 2025; Li et al., 2025b). None repairs the realized flips of a shipped artifact in its serving format under a byte and latency budget, or declares the slices it leaves out of tolerance.

7 Conclusion

Quantization damage in an LLM concentrates on the slices held to the strictest tolerance, which aggregate metrics hide. QGATE repairs a shipped quantized model by treating its flips as counterexamples, patching the tolerance-normalized worst slice first within a byte and latency budget, and sealing a per-slice report. At matched cost, where the budget goes decides the outcome: on cfg1 QGATE cuts the worst slice by 37.2% against 23.0% for an aggregate-loss search over the same patches, and on cfg2 a summed objective reaches 42.5% against 60.3%. A quantized checkpoint can thus be repaired in its shipped format, with a report that states which slices remain uncertified.

REFERENCES

Muhammad Alif Al Hakim, Alfan Farizki Wicaksono, and Fajri Koto. Preserving fairness and safety in quantized LLMs through critical weight protection. In *Findings of the Association for Computational Linguistics: ACL*, 2026. doi: 10.18653/v1/2026.findings-acl.993.

Aymen Bouguerra, Daniel Alfonso Montoya Vasquez, Alexandra Gomez-Villa, Chokri Mraidha, and Fabio Arnez. Less precise can be more reliable: A systematic evaluation of quantization's impact on VLMs beyond accuracy. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Wvd80GPJ9F.

Ting-Yun Chang, Muru Zhang, Jesse Thomason, and Robin Jia. Why do some inputs break low-bit LLM quantization? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. doi: 10.18653/v1/2025.emnlp-main.168.

Peter Clark et al. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Karl Cobbe et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Alexis Conneau et al. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Devleena Das, Rajeev Patwari, and Ashish Sirasao. Recover-LoRA: Data-free accuracy recovery of degraded language models via low-rank adaptation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 2025. doi: 10.18653/v1/2025.emnlp-industry.164.

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 *International Conference on Learning Representations*, 2024.

Abhinav Dutta, Sanjeev Krishnan, Nipun Kwatra, and Ramachandran Ramjee. Accuracy is not all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Elias Frantar et al. Gptq: Accurate post-training quantization for generative pre-trained transformers, 2022.

Georgi Gerganov and llama.cpp contributors. ggml-org/llama.cpp: LLM inference in C/C++. github.com, 2024. URL https://github.com/ggml-org/llama.cpp.

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

Meraj Hashemizadeh, Juan Ramirez, Rohan Sukumaran, Golnoosh Farnadi, Simon Lacoste-Julien, and Jose Gallego-Posada. Balancing act: Constraining disparate impact in sparse models. In *International Conference on Learning Representations*, 2024.

Dan Hendrycks et al. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

Haoyu Huang, Linlin Yang, Sheng Xu, Boyu Liu, Guodong Guo, Zhongqian Fu, Hang Zhou, and Baochang Zhang. FAIR-Calib: Frontier-aware instability-reweighted calibration for post-training quantization of diffusion large language models. arXiv preprint arXiv:2606.06547, 2026.

Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi. SliM-LLM: Salience-driven mixed-precision quantization for large language models. In *International Conference on Machine Learning*, 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 *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Alex Kogan. Is (selective) round-to-nearest quantization all you need?, 2025. URL https://arxiv.org/abs/2505.15909.

Jason Kong et al. A kl lens on quantization: Fast, forward-only sensitivity for mixed-precision ssm-transformer models, 2026.

Jonas Kübler et al. When llms get significantly worse: A statistical approach to detect model degradations, 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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. doi: 10.1609/aaai.v38i12.29237.

Jemin Lee, Sihyeong Park, Jinse Kwon, Jihun Oh, and Yongin Kwon. Exploring the trade-offs: Quantization methods, task difficulty, and model size in large language models from edge to giant. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2025. doi: 10.24963/ijcai.2025/902.

Amit LeVi et al. You had one job: Per-task quantization using llms' hidden representations, 2025.

Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. GPTAQ: Efficient finetuning-free quantization for asymmetric calibration. In *International Conference on Machine Learning*, 2025a.

Zhen Li, Yupeng Su, Songmiao Wang, Runming Yang, Congkai Xie, Aofan Liu, Ming Li, Jiannong Cao, Yuan Xie, Ngai Wong, and Hongxia Yang. Quantization meets reasoning: Exploring and mitigating degradation of low-bit LLMs in mathematical reasoning. arXiv preprint arXiv:2505.11574, 2025b.

Ji Lin et al. Awq: Activation-aware weight quantization for llm compression and acceleration, 2023.

Hongyuan Liu et al. Spear: A system for post-quantization error-adaptive recovery enabling efficient low-bit llm serving, 2026.

Jiawei Liu, Lin Niu, Zhihang Yuan, Dawei Yang, Xinggang Wang, and Wenyu Liu. PD-Quant: Post-training quantization based on prediction difference metric. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. doi: 10.1109/cvpr52729.2023.02340.

João Batista P. Matos Jr. et al. Ceg4n: Counter-example guided neural network quantization refinement, 2022.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Alicia Parrish et al. BBQ: A hand-built bias benchmark for question answering. In *Findings of the Association for Computational Linguistics: ACL 2022*, 2022.

Qwen, 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, Tianyi Tang, 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. arXiv preprint arXiv:2412.15115, 2024.

Baha Rababah, Shahzeb Qamar, Lorenz Sparrenberg, Rafet Sifa, Murat Kantarcioglu, Cuneyt Gurcan Akcora, and Carson K. Leung. The illusion of equivalency: Statistical characterization of quantization effects in LLMs. arXiv preprint arXiv:2607.08734, 2026.

Plawan Kumar Rath et al. Quantization undoes alignment: Bias emergence in compressed llms across models and precision levels, 2026.

Keisuke Sakaguchi et al. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Oliver Sieberling, Denis Kuznedelev, Eldar Kurtic, and Dan Alistarh. EvoPress: Accurate dynamic model compression via evolutionary search. In *International Conference on Machine Learning*, 2025.

Xidan Song et al. Qnnrepair: Quantized neural network repair, 2023.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL*, 2023. doi: 10.18653/v1/2023.findings-acl.824.

Qitao Tan et al. Q-realign: Piggybacking realignment on quantization for safe and efficient llm deployment, 2026.

turboderp. turboderp-org/exllamav2: A fast inference library for running LLMs locally on modern consumer-class GPUs. github.com, 2024. URL https://github.com/turboderp-org/exllamav2.

Sunghyun Wee, Suyoung Kim, Hyeonjin Kim, Kyomin Hwang, and Nojun Kwak. Safety-preserving PTQ via contrastive alignment loss. arXiv preprint arXiv:2511.07842, 2025.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2018.

Hanqi Xiao et al. Task-circuit quantization: Leveraging knowledge localization and interpretability for compression, 2025.

Rowan Zellers et al. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the Association for Computational Linguistics (ACL)*, 2019.

Cheng Zhang, Jeffrey T. H. Wong, Can Xiao, George A. Constantinides, and Yiren Zhao. Qera: an analytical framework for quantization error reconstruction. In *International Conference on Learning Representations (ICLR)*, 2025.

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 *Findings of the Association for Computational Linguistics: ACL*, 2026. doi: 10.18653/v1/2026.findings-acl.1162.

AI USE STATEMENT

We used generative AI tools to translate the text (Claude Opus 5.0), to check grammar and spelling and polish the wording (GPT-5.6), to guide the drawing of the figures (Gemini), and to check the content (Grok). We did not use generative AI tools to modify code. We have reviewed all AI-assisted work: we checked every reference for hallucinated citations. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The details needed to reproduce the results of this paper, including the frozen hyperparameters, the repair loop and its conventions, the sealed sampling rule, the evaluation protocol and prompts, and the software and hardware setup, are disclosed in this appendix.

APPENDIX

Table 6 lists the contents of each part of the appendix.

Table 6: What each part of the appendix holds.

Section	Contents
Appendix A.0	Limitations
Appendix A.1	Extended related work
Appendix A.2	Theory and multiplicity
Appendix A.3	Method details, per-slice reports, and the Step-0 census
Appendix A.4	Additional experimental results
Appendix A.5	Compute, baselines, and latency
Appendix A.6	Reproducibility and hyperparameters
Appendix A.7	Motivation panels in full

APPENDIX A.0: LIMITATIONS

Budget sensitivity. A deployer who picks a byte budget needs to know how many slices would clear their tolerance on the discovery data if that budget were tighter or looser, and how closely the latency surcharge model predicts the measured decode latency at each budget. The paper lacks this budget study for the primary configuration.

Per-slice headroom. The margin by which a certified slice clears its tolerance tells a deployer how much further drift the shipped artifact can absorb before that slice fails. The paper lacks an analysis of this headroom across the certified slices of the primary configuration.

Budget-response curve. A curve of the worst-slice reduction against the byte budget, drawn for the flip-neighborhood shortlist and for a random candidate pool side by side, shows a deployer at a glance where extra bytes stop paying off. The paper lacks such a curve.

APPENDIX A.1: EXTENDED RELATED WORK

Quantization backbones and precision allocation. Post-training quantization of LLMs rests on a small set of backbones. GPTQ (Frantar et al., 2022) solves a layer-wise reconstruction with approximate second-order information, GPTAQ (Li et al., 2025a) reconstructs quantized-layer outputs against full-precision ones under asymmetric calibration, SpQR (Dettmers et al., 2024) and SqueezeLLM (Kim et al., 2024) isolate outlier weights in a sparse high-precision part, and Selective RTN (Kogan, 2025) revisits round-to-nearest. A second family allocates precision non-uniformly: AWQ (Lin et al., 2023) and OWQ (Lee et al., 2024) protect salient weights and channels, SliM-LLM (Huang et al., 2025) assigns group-wise mixed precision by salience, sensitivity-based methods allocate precision from forward or gradient sensitivity scores (Kong et al., 2026; LeVi et al., 2025; Xiao et al., 2025), and EvoPress (Sieberling et al., 2025) searches per-layer profiles against an aggregate objective. Serving formats such as EXL2 (turboderp, 2024) and GGUF (Gerganov & llama.cpp contributors, 2024) fix what a patch may change. All of these methods decide precision before any flip is observed; they produce the base artifact that QGATE then patches, so the two compose rather than compete.

Behavioral effects of quantization and their detection. A growing body of evidence shows that quantization moves behavior that aggregate metrics average away. Base and quantized models are behaviorally non-equivalent under matched scores (Rababah et al., 2026); quantization partially undoes alignment and re-introduces demographic bias at specific precision levels (Rath et al., 2026); and the change is heterogeneous across task difficulty, model size, and robustness type, in language and vision-language models alike (Lee et al., 2025; Bouguerra et al., 2026). Failure-localization studies trace the damage to signal degradation and computation collapse and to the inputs that break low-bit models (Zhou et al., 2026; Chang et al., 2025), argue that flips rather than accuracy are the

quantity to track (Dutta et al., 2024), and paired statistical tests detect degradations too small for a raw score comparison (Kübler et al., 2026). These works establish where and whether a quantized model is damaged; none prices a repair of the damage or reports what a repaired artifact still leaves unfixed, which is the gap QGATE's ledger, loop, and report fill in that order.

Fairness- and safety-preserving compression. Several methods protect a chosen behavior during compression. Balancing Act (Hashemizadeh et al., 2024) constrains per-group accuracy change during sparsification, FAIR-Calib (Huang et al., 2026) reweights a hidden-state calibration objective, Critical Weight Protection (Al Hakim et al., 2026) preserves the weights that carry fairness and safety, Q-realign (Tan et al., 2026) piggybacks safety realignment on quantization, and a contrastive-alignment loss keeps safety behavior through post-training quantization (Wee et al., 2025). Each fixes a behavior class in advance and optimizes it during compression or training. QGATE instead reads the realized flips of the shipped artifact, and its worst-case objective makes the protection non-substitutable: because acceptance requires a lexicographic improvement of the sorted relative-flip vector, a patch cannot buy the targeted slice with a regression on the worst coordinate, and any collateral movement it permits on a lower coordinate is charged against the candidate on the swept ledger rather than ignored.

Post-hoc repair and accuracy recovery. Repair after quantization has taken three forms. Counterexample-guided refinement (Matos Jr. et al., 2022) and quantized-network repair (Song et al., 2023) use exact solvers and inherit their soundness, which confines them to classifiers orders of magnitude smaller than an LLM; QGATE's verifier is an empirical full-ledger sweep, trading soundness on a small network for feasibility on a large one. Compensators and adapters recover accuracy by adding error-adaptive modules or low-rank adapters at serving time (Liu et al., 2026; Das et al., 2025; Zhang et al., 2025), and PD-Quant (Liu et al., 2023) acts at calibration time on a divergence between full-precision and quantized predictions. Tuned recovery restores a degraded capability such as mathematical reasoning (Li et al., 2025b). Keeping the patch inside the existing kernel format, rather than adding modules or tuning, is what keeps QGATE's shipped artifact a drop-in replacement whose deployment cost is expressible in bytes and scored latency alone.

Lineage of the counterexample loop. The outer shape of QGATE, propose a repair, check it against the full set of known counterexamples, fold any newly discovered counterexample back in, and repeat, is the counterexample-guided pattern shared by inductive synthesis and abstraction refinement. In that classical setting the checker is an exact oracle (a solver or model checker) that either certifies the candidate or returns a concrete counterexample, and the loop inherits the oracle's soundness. QGATE keeps the loop but cannot keep the oracle: no exact behavioral-equivalence oracle exists for a 7B-parameter transformer beyond a counted universe of natural-language slices. Two design choices follow directly. First, the "counterexamples" are realized flips against the FP reference rather than solver-produced witnesses, so the discovery ledger plays the role the oracle's counterexample set plays classically, and the canary tier detects flips a repair creates during trials. Second, acceptance is a full-ledger empirical sweep rather than a soundness proof.

APPENDIX A.2: THEORY AND MULTIPLICITY

A.2.1 SCOPE OF THE REPORTED QUANTITIES

A regime-E slice's universe is the entire standard evaluation split, the same finite set the discovery ledger is built on, so there is no pool left over from which a disjoint sealed draw could be taken. What we report is instead a full-census measurement of the shipped artifact on that whole universe: a count of disagreeing items and the proportion it induces, with no sampling error, no interval, and no confidence budget consumed, because nothing is being extrapolated.

A.2.2 WHY AN EXHAUSTIVELY EVALUATED SLICE CARRIES NO INTERVAL

On a regime-E slice the reported value is a full-census count over the entire universe, so no Clopper-Pearson interval is attached, because an interval would answer a sampling question these slices do not pose, and they consume no confidence budget. A paired McNemar comparison (McNemar, 1947) reported on such a slice is a different matter, since it is an inferential statement about which

of two artifacts is better on discordant items, and it does spend from the separate paired budget. The two budgets are therefore disjoint by construction: an exhaustive slice never spends from the Clopper-Pearson budget.

Theorem 1 (Residual bound)**.** *Let each shipped artifact $\hat{q}$ (one per configuration) be returned by the procedure of Algorithm 1 using only $\mathcal{D}_{\text{disc}}$ and frozen with all shipped choices before any sealed data is seen. For each shipped artifact and each regime-L slice k draw n_k i.i.d. inputs from S_k, sealed and disjoint from discovery data, with empirical flip rate $\hat{F}_k$. Then with probability at least $1-\alpha_{\text{CP}}$, simultaneously over all shipped artifacts and their regime-L slices (the $3 \times K_L$ enumerated rows that share the $\alpha_{\text{CP}} = 0.03$ budget), the bound of Eq. (18) holds, with α'_k the Bonferroni share from the pre-specified allocation table below.*

Two hypotheses are required: each sealed size n_k is fixed before any sealed data is seen, and the registered family over which the shares α'_k are allocated is likewise fixed before unsealing. Neither may be adjusted after the pool is opened.

Proof. Because $\hat{q}$ and every shipped choice (budget point, hyperparameters, slice specs, tolerances) are fixed functions of $\mathcal{D}_{\text{disc}}$, the sealed samples drawn afterward are i.i.d. conditional on those choices. For each regime-L slice k, $\hat{F}_k n_k \sim \text{Binomial}(n_k, F_k(\hat{q}))$, and the one-sided Clopper-Pearson upper limit at level α'_k satisfies $\Pr[F_k(\hat{q}) \leq \text{CP}_{\text{upper}}(\hat{F}_k, n_k, \alpha'_k)] \geq 1 - \alpha'_k$ by the exact beta-binomial inversion. Applying a Bonferroni union bound over the enumerated regime-L rows gives simultaneous coverage $1 - \alpha_{\text{CP}}$. □

Remark 2 (Scope). Three scope conditions travel with Theorem 1 and are not part of its mathematical content. First, the statement is scoped to regime-L slices and to them alone: a regime-E slice is not covered by it and needs no coverage, since its reported number is a full-census count over the entire universe, so there is no estimand separate from the measurement and no interval to guarantee, and it consumes no α; any paired McNemar test reported on such a slice is a separate inferential row and still spends a share of the α_{pair} budget. Second, the bound certifies only smallness and never degradation. Third, validity is conditional on i.i.d. sampling from the slice distributions, covers the shipped model rather than Eq. (1) optimality, and does not equate FP agreement with correctness.

Power table. The sample sizes are fixed before unsealing so that a genuinely repaired slice is very likely to certify. The design point is a true rate at or below $\tau_k/2$: a slice the loop has pulled to half its tolerance should pass its one-sided bound with high probability, whereas a slice still near τ_k sits near the certification boundary and may ship inconclusive. The floor is a property of that design point alone: certification probability falls monotonically as the true rate rises from $\tau_k/2$ toward τ_k. Against a floor of 0.95 certification probability at the $\tau_k/2$ design point, at the allocated per-bound share $0.03/(3K_L) = 2 \times 10^{-3}$, the registered sizes are $n_k \approx 3300$ at $\tau_k = 0.02$ and $n_k \approx 1400$ at $\tau_k = 0.05$. They are not the smallest that clear that floor, so each carries margin above it: a larger n_k only tightens the bound. The choice of $\tau_k/2$ as the detectable rate is what couples the sizing to the tolerances rather than to the observed rates, keeping the draw pre-registration-safe. All regime-L slices in the manifest are $\tau_k = 0.05$ cells drawn at $n_k = 1400$.

Multiplicity allocation. Table 7 is the one pre-specified allocation. Every unsealed interval or test reported in this paper is one of these 62 rows; no max-over-slices inferential claim is made on sealed data. A regime-E slice's residual rate is exact and needs no CP row, but any paired McNemar comparison reported on it is one of the enumerated α_{pair} rows. The rows the reader can audit against the tables are these: the 15 α_{CP} rows are exactly the 3 shipped configs $\times$ 5 regime-L slices of Tables 1, 11, and 12; within the α_{pair} budget, the paper prints 11 baseline-versus-QGATE McNemar tests (Table 3), while the remaining registered paired rows spend their share whether or not a p-value is printed: the 5 degenerate regime-L rows, the 6 per-config baseline comparisons of Table 15, the 18 regime-E paired comparisons whose outcome Table 8 reports as an exact delta rather than a p-value, and 7 further comparisons registered as reserve and never exercised, which accounts for all 47.

Why the allocation is fixed before unsealing. The single-unseal discipline is what lets a union bound carry the whole report. All 62 rows, their tests, and their per-row shares are registered while only the discovery data is visible; the sealed pool is then opened exactly once, every registered bound is computed, and nothing further is tested. The family-wise statement is over the registered

family, so it holds for the printed subset enumerated above without any further correction. Because the allocation is data-independent of the sealed draw, the Bonferroni union over the regime-L rows is valid without any post-hoc correction, and the reported per-slice bounds hold simultaneously at the stated overall confidence rather than marginally one slice at a time.

The fail-closed decision. Certification is a deterministic function of the sealed bound. A regime-L slice is certified iff its one-sided Clopper-Pearson upper limit at the allocated share lies at or below the slice tolerance τ_k; a regime-E slice is certified iff its exact rate does. Any slice failing its test is not retried, re-sampled, or dropped: it is written into the shipped report as an explicit uncertified line carrying its reported residual value, the exact full-census rate for a regime-E slice and the Clopper-Pearson upper bound for a regime-L one, so a slice that cannot be certified within the budget is visible rather than absent. The rule never conditions on the sealed outcome to change a shipped choice, which is what keeps the coverage statement of the residual bound intact.

Table 7: The 62-row α allocation ($\alpha = 0.04$). The paired share is $0.01/47 = 2.13 \times 10^{-4}$, the 47 rows being 11 printed baseline tests, 5 degenerate regime-L rows, 6 per-config baseline comparisons, 18 regime-E paired comparisons, and 7 reserve rows.

Group	α	rows	per-row share
α_{CP} (regime-L CP bounds)	0.03	15	2.0e-3
α_{pair} (paired McNemar)	0.01	47	2.13e-4
Total	0.04	62	–

A.2.3 Elementary properties of the grammar, the loop, and the allocation

The main text uses the following five facts without argument.

Lemma 3 (P1 is idempotent and monotone in bits)**.** *Let* GROUP-RAISE$(l, \mathcal{G})$ *act on each $g \in \mathcal{G}$ by Eq. (6). Then $b_{\mathrm{new}}(g) \geq b_{\mathrm{old}}(g)$ for every g, and applying the same instance twice leaves the artifact unchanged.*

Proof. $b_{\mathrm{new}}(g) = \max(b_{\mathrm{old}}(g), \mathrm{target}) \geq b_{\mathrm{old}}(g)$ gives monotonicity. Re-applying with the same target maps $b_{\mathrm{new}}(g)$ to $\max(b_{\mathrm{new}}(g), \mathrm{target}) = b_{\mathrm{new}}(g)$, since $b_{\mathrm{new}}(g) \geq \mathrm{target}$ already. The target is absolute rather than incremental, so the second application is a no-op. □

Proposition 4 (Budget feasibility is an invariant of the accepted sequence)**.** *Every prefix A_t of the returned sequence $\hat{A}$ satisfies* $\mathrm{bytes}(A_t) \leq B$ *and* $\mathrm{lat}(A_t) \leq L$*, and both quantities are non-decreasing in t.*

Proof. By Lemma 3 each P1 instance has $b_{\mathrm{new}}(g) - b_{\mathrm{old}}(g) \geq 0$, so its cost in Eq. (7) is non-negative; P2 and P3 rewrite no weights and cost zero. Hence bytes is a sum of non-negative terms and is non-decreasing along the sequence. The base case $\mathrm{bytes}(()) = 0 \leq B$ holds, and the acceptance test admits a only when $\mathrm{bytes}(A_t) + \Delta\mathrm{bytes}(a) \leq B$, which is the inductive step. The same argument applies verbatim to the latency cap: every entry of the frozen surcharge table is a non-negative delta, so lat is likewise a sum of non-negative terms and non-decreasing along the sequence, the base case $\mathrm{lat}(()) = 0 \leq L$ holds, and the acceptance test admits a only when $\mathrm{lat}(A_t) + \mathrm{surch}(\mathrm{op}(a), \mathrm{gran}(a)) \leq L$. Hence every prefix satisfies $\mathrm{lat}(A_t) \leq L$ as well. □

Proposition 5 (Finite termination)**.** *The loop of Algorithm 1 performs finitely many accepted steps and halts.*

Proof. For slice k the discovery ledger is a fixed finite set of size M_k, so $\hat{F}_{k,\mathrm{disc}}$ takes values in the finite set $\{0, 1/M_k, \ldots, 1\}$ and $\hat{F}_{k,\mathrm{disc}}/\tau_k$ likewise takes finitely many values. Therefore V_{disc} of Eq. (13), a sorted tuple of K such coordinates, ranges over a finite set. Acceptance requires strict lexicographic decrease of V_{disc} (Eq. (14)) verified on the full ledger, and the lexicographic order on a finite set is well-founded, so no infinite strictly decreasing chain exists and only finitely many acceptances can occur. Each step scores a finite shortlist of the frozen grammar and performs at most $S_{\max}$ sweeps, and a step with no acceptance exits the loop, so the procedure halts. □

Lemma 6 (Registering a superset is conservative)**.** *Let the registered family have R rows and let the executed subset have $E \leq R$ rows, all shares equal within a group. Then the registered per-row share satisfies $\alpha/R \leq \alpha/E$.*

Proof. $E \leq R$ and $\alpha > 0$ give $\alpha/R \leq \alpha/E$. A smaller per-row share yields a wider one-sided interval and a stricter p-value threshold, so every registered row is tested at least as conservatively as it would be under an allocation over the executed rows alone. □

Coverage across the two groups. The allocation of Table 7 is split into two groups, and the overall statement is a union bound over the split rather than over a single list. Within the Clopper-Pearson group, a Bonferroni union over its rows gives simultaneous coverage at least $1 - \alpha_{\mathrm{CP}}$; within the paired group, the same argument gives family-wise error at most α_{pair}. A further union bound across the two groups gives an overall family-wise error at most $\alpha_{\mathrm{CP}} + \alpha_{\mathrm{pair}} = \alpha$ over the registered family, which is the joint statement the report makes. The allocation controls multiplicity across that family; it does not convert an in-sample paired row into an out-of-sample one, and the paired rows are read as the in-sample comparisons Appendix A.4.2 says they are. No independence between the groups is assumed anywhere, which is why the union bound and not a product is used.

APPENDIX A.3: METHOD DETAILS

A.3.1 PER-SLICE REPORTS

Selectivity of the repair. Table 8 reads the paired comparison cell by cell to test where the loop actually acted. Movement is confined to the families that carried the flips: the nine bias and safety cells above $\tau = 0.02$ move, with per-cell reductions from 0.0138 (BBQ-SES) down to 0.0026 (BBQ-Nationality), while BBQ-GenderId, which starts just under tolerance at ratio 0.99, moves 0.0014 as a spillover, and the eight remaining regime-E cells register a zero exact delta while all five regime-L cells carry a degenerate (undefined) paired test. The reductions track initial violation only loosely, which we attribute to re-ranking: the loop stops working a cell as soon as it ceases to be the relative worst, so BBQ-SES, which started second, absorbs the largest reduction while BBQ-PhysAppear, which started just behind it, absorbs less. Confinement is measured rather than guaranteed, and it holds at family granularity rather than cell granularity, since BBQ-GenderId is pulled down as a spillover of patches the loop applied to its neighbours.

A.3.2 BUDGET SWEEP, TABULATED

Impact of budget and shortlist policy. Table 9 sweeps the byte budget on cfg2 and contrasts the flip-neighborhood shortlist against a uniform-random candidate pool. The shortlist reaches its full reduction by 96 MB while the random pool plateaus below it, at a final gap of 6.9 points; both are flat at 120 MB, the shortlist because its worst slice passed tolerance and the loop halted, the random pool because it exhausted its candidates with its worst slice still at 0.022. The two rows stay within about three points in the budget-starved regime, where any reasonable candidate closes the largest violations, so the advantage lies in the tail: once the obvious repairs are spent, we attribute the tail gap to the shortlist concentrating candidates on the flip neighborhood, which a uniform pool covers only by chance. Each row is a single-trajectory prefix, not a re-run under each cap. The random pool's plateau value follows from its own reduction: a worst slice of 0.0469 cut by 53.4% leaves 0.0219, the 0.022 quoted above.

A.3.3 CONVENTIONS OF THE GRAMMAR AND THE LOOP

The vectors $h_l^{f_0}(x)$ and $h_l^{q}(x)$ live in the same layer-l output space and are compared entrywise.

Since $K^\star$ is a set of slice indices rather than a single one, the ledger it indexes is the union over its members, $\mathcal{D}_{\mathrm{disc},K^\star} := \bigcup_{k \in K^\star} \mathcal{D}_{\mathrm{disc},k}$, which reduces to the single-slice ledger $\mathcal{D}_{\mathrm{disc},k}$ when the maximizer is unique. The input set carries no layer index: the same flipped items are read at every layer, and only the traffic statistic traf_l is layer-specific.

Candidates are cut from each projection matrix as contiguous output-channel ranges of one fixed width.

Table 8: Paired comparison $\hat{q}$ vs q_0 (cfg1, full manifest). For regime-E rows the rates are exact on the full universe and $\Delta = q_0 - \hat{q}$; regime-L rows give the paired McNemar result, where n.s. marks a degenerate rather than a merely non-significant test: the loop left these cells unchanged, so there are no discordant pairs and the McNemar statistic is undefined; the row still occupies its registered α_{pair} share. A zero Δ is a decision-level statement at the reported four-decimal resolution: it says the flip count of that universe is unchanged. The ten cells the loop moves carry Δ of .0138, .0114, .0103, .0102, .0069, .0065, .0051, .0050, .0026, and .0014, nine above-tolerance cells and one same-family spillover, with a median of .0067.

Slice	Regime	q_0 rate	$\hat{q}$ rate	Δ / p
BBQ-SES	E	.0287	.0149	.0138 (exact)
BBQ-Age	E	.0307	.0193	.0114 (exact)
BBQ-Disability	E	.0244	.0141	.0103 (exact)
BBQ-PhysAppear	E	.0286	.0184	.0102 (exact)
BBQ-SexOrient	E	.0243	.0174	.0069 (exact)
Safety-Refusal	E	.0245	.0180	.0065 (exact)
BBQ-RaceEth	E	.0202	.0151	.0051 (exact)
BBQ-Religion	E	.0242	.0192	.0050 (exact)
BBQ-Nationality	E	.0214	.0188	.0026 (exact)
BBQ-GenderId	E	.0197	.0183	.0014 (exact)
MMLU-STEM	E	.0209	.0209	.0000 (exact)
MMLU-Humanities	E	.0179	.0179	.0000 (exact)
MMLU-SocialSci	E	.0296	.0296	.0000 (exact)
MMLU-Other	E	.0233	.0233	.0000 (exact)
ARC-Challenge	E	.0427	.0427	.0000 (exact)
BBH-fc	E	.0363	.0363	.0000 (exact)
WinoGrande	E	.0205	.0205	.0000 (exact)
GSM8K-MC	E	.0243	.0243	.0000 (exact)
HellaSwag	L	.0150	.0150	n.s.
XNLI-es	L	.0171	.0171	n.s.
XNLI-fr	L	.0243	.0243	n.s.
XNLI-de	L	.0200	.0200	n.s.
XNLI-sw	L	.0271	.0271	n.s.

Table 9: A3 budget/pool ablation on cfg2 (worst-slice reduction %). Columns are budget caps, not bytes spent; the shipped cfg2 run spends 80 MB of the 96 MB cap.

Budget (MB)	12	24	48	72	96	120
shortlist (ours)	31.5	38.4	50.7	58.9	60.3	60.3
random pool	30.1	35.6	49.3	53.4	53.4	53.4

The unit that the format addresses and the unit that the loop accepts are distinct, and the byte accounting follows the latter. A native group of $|g|$ weights is the granularity at which a bit-width field is stored; an action instance is a batch of them. The group family $\mathcal{G}$ of one P1 instance is the set of native groups covered by the shortlisted tensor slice, a contiguous range of output channels of one projection matrix of layer l, which is the region over which the flip-neighborhood traffic statistic is resolved and the region a re-emission rewrites as a unit.

The repetition and conflict rules are that a region may be raised at most once per sequence, that a later P1 instance may not overlap an accepted one, and that P2 and P3 may each fire at most once per layer and no P1 may follow them on that layer. P3 is enabled per layer class from what the loader exposes for the artifact actually loaded, so its availability is read off that checkpoint rather than assumed from the architecture; a class with no exposed bias term simply admits no P3 instance, and the zero-byte family reduces to P2 on the classes where that is the case.

A tied worst coordinate may be reduced without raising the others, so ties alone never block the acceptance test.

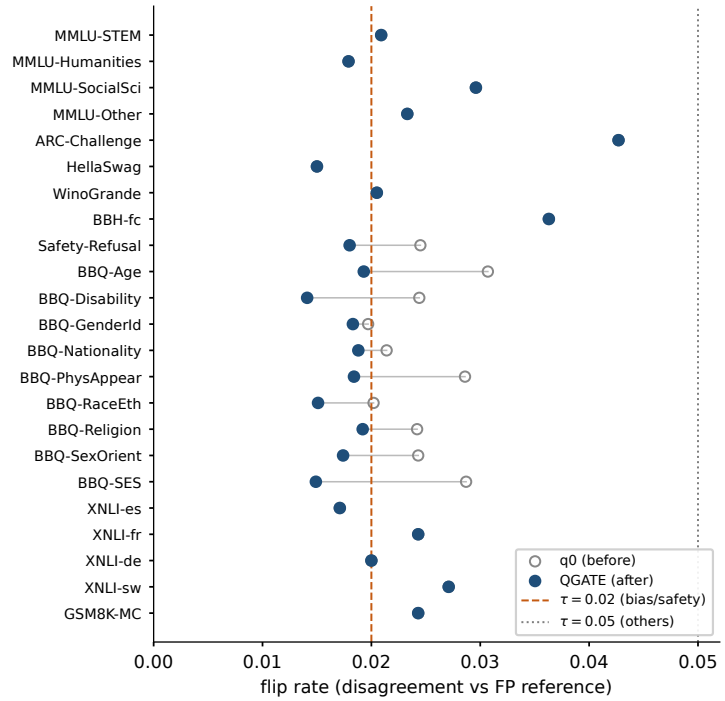


Figure 4: Per-slice flip rate before (q0) and after QGATE on cfg1. Bias and safety cells above $\tau = 0.02$ are pulled under tolerance; already-passing regime-L cells are left untouched, while the one already-passing bias cell moves as a same-family spillover of the patches applied to its neighbours.

The surrogate vector and surrogate potential are the constructions of Eq. (13) and Eq. (16) with $\tilde{F}_k$ substituted for $\hat{F}_{k,\mathrm{disc}}$,

$$\tilde{V}(q) = \mathrm{sort}_{\downarrow}\big(\tilde{F}_1(q)/\tau_1, \ldots, \tilde{F}_K(q)/\tau_K\big), \tag{19}$$

which is the object the surrogate acceptance test is read on, and whose companion potential aggregates the same relative rates into the scalar that ranks the candidates surviving that test,

$$\tilde{\Phi}(q) = \sum_{k=1}^{K} \frac{\tilde{F}_k(q)}{\tau_k}. \tag{20}$$

A candidate failing its sweep is rejected and the next-ranked one swept, up to a cap of $S_{\max}$ sweeps per step. A byte-positive P1 candidate that no longer fits the remaining budget is pruned, while zero-byte P2/P3 candidates within the latency cap remain eligible.

A slice is uncertified in either of two ways: inconclusive (empirical rate at or below tolerance but bound above it) and fail-closed-unrepaired (rate above tolerance). The fail-closed counts of Table 16 are the narrower fail-closed-unrepaired class alone, with the inconclusive slices listed in the per-config sealed reports.

Algorithm 1 states the loop with the conventions of this subsection folded in. Each step reads the tolerance-normalized worst set $K^{\star}$ off the current artifact, resolves the flip neighborhood on the ledger that set indexes, and shortlists grammar instances by the traffic they carry there. Each shortlisted instance is scored on the active and canary tiers alone and enters the ranking only if the surrogate vector improves, the zero-byte class ranked first by surrogate potential drop and the byte-positive class after it by drop per byte; ranked instances the repetition and conflict rules exclude, and those whose bytes or scored latency would carry the accepted sequence past B or L, are dropped before any sweep, so an infeasible candidate never consumes one. Ranking expresses a preference among admitted candidates rather than a second admission test: the full-ledger sweep of $\mathcal{D}_{\mathrm{disc}}$ is what commits an action, and a candidate failing it is discarded in favor of the next ranked one, up

Algorithm 1 QGATE Counterexample-Guided Repair

Input: base artifact q_0, discovery ledger $\mathcal{D}_{\text{disc}}$ with its active and canary tiers, grammar G, tolerances $\{\tau_k\}$, byte budget B, latency cap L, sweep cap $S_{\max}$
Output: patch sequence $\hat{A}$, frozen artifact $\hat{q}$, sealed per-slice report
Initialize $q \leftarrow q_0$, $A \leftarrow ()$, accepted steps $\leftarrow 0$
while $\max_k \hat{F}_{k,\text{disc}}(q)/\tau_k > 1$ **and** accepted steps < 16 **do**
  $K^\star \leftarrow \arg\max_k \hat{F}_{k,\text{disc}}(q)/\tau_k$ (all tied maximizers); read the flip neighborhood $\mathcal{N}$ of $K^\star$ at q and shortlist the instances of G carrying the largest traf_l on it
  Score each candidate a on the two tiers at $q' = T(q, a, \mathcal{D}_{\text{disc}})$ and retain it only if $\tilde{V}(q') <_{\text{lex}} \tilde{V}(q)$ via Eq. (19)
  Rank the retained instances as one list, zero-byte ones first by $\tilde{\Phi}(q) - \tilde{\Phi}(q')$ of Eq. (20), then byte-positive ones by $\rho(q, q')$ of Eq. (17); set accepted $\leftarrow$ **false**
  Prune every ranked a excluded by the repetition and conflict rules, or with $\text{bytes}(A) + \Delta\text{bytes}(a) > B$, or with $\text{lat}(A) + \text{surch}(\text{op}(a), \text{gran}(a)) > L$
  for each a in ranked order, at most $S_{\max}$ sweeps **do**
    $q' \leftarrow T(q, a, \mathcal{D}_{\text{disc}})$ via Eq. (12); sweep the full ledger $\mathcal{D}_{\text{disc}}$ at q'
    if $V_{\text{disc}}(q') <_{\text{lex}} V_{\text{disc}}(q)$ **then**
      Set $q \leftarrow q'$, append a to A, increment accepted steps, re-read the active tier at q with the canary draw held fixed, set accepted $\leftarrow$ **true**, and **break** (Eq. (14))
    end if
  end for
  if accepted is **false then break** (a rejected trial spends a sweep, never a step)
end while
// Phase boundary: the artifact is frozen; nothing below changes it
Set $\hat{A} \leftarrow A$ and $\hat{q} \leftarrow q_{\hat{A}}$, and freeze every shipped choice
Open the sealed pools once and read $\hat{q}$ on them, emitting per slice its bound or exact rate, its tolerance τ_k, and its status
return $\hat{A}$, $\hat{q}$, and the sealed per-slice report

to $S_{\max}$ sweeps. Only a commitment advances the state and the accepted-step counter, so rejected trials spend sweep effort and never the step cap. The loop exits when every slice is under tolerance on the discovery ledger, when a step commits nothing, or at the step cap, whichever comes first, and the last lines mark the phase boundary: the artifact is frozen, and only then are the sealed pools opened once to produce the per-slice status.

A.3.4 The Step-0 census read cell by cell

How to read a printed rate. In Table 1, regime L is Clopper-Pearson sampling ($n_k < N_k$) and regime E is exhaustive-exact ($n_k = N_k$); N_k names the certified universe on a regime-E row and the label-free sampling pool on a regime-L row, whose estimand is the flip proportion in that pool. Every census and sealed-report rate in this paper is a count over the corresponding slice universe, the exhaustive split for a regime-E slice and the $n_k = 1400$ sealed draw for a regime-L one, so a reader can invert any printed rate to the integer count behind it. The ablation tables are the one exception: their rates are counts over the discovery ledger of size M_k. Percentage reductions are computed from those integer counts rather than from the four-decimal rates shown, so they may differ slightly from the naive ratio of the displayed rates. Each regime-L row inverts to a flip count x out of $n_k = 1400$, and its printed bound is the one-sided Clopper-Pearson upper limit $\text{BetaInv}(1 - \alpha'_k;\, x+1,\, n_k - x)$ evaluated at $\alpha'_k = 2 \times 10^{-3}$. The fifteen counts, in the slice order HellaSwag, XNLI-es, XNLI-fr, XNLI-de, XNLI-sw, are 21, 24, 34, 28, 38 on cfg1; 38, 41, 53, 42, 49 on cfg2; and 41, 43, 38, 37, 50 on cfg3. Those fifteen counts read against a single threshold: at $n_k = 1400$ and $\alpha'_k = 2 \times 10^{-3}$ the largest count that still certifies is 47, whose bound is 0.0498, while 48 gives 0.0507, so a regime-L row certifies exactly when its sealed count is at most 47, a point rate of 0.0336. Twelve of the fifteen sit at or below that threshold and carry a passing bound; the three that stand above it, 53 and 49 on cfg2 and 50 on cfg3, are the same three rows the sealed reports mark inconclusive. The three worst-slice reductions invert the same way: BBQ-Age on cfg1 moves 113 disagreeing items to 71 out of $N_k = 3680$, giving 37.2%; BBQ-Disability on cfg2 moves 73 to 29 out of 1556, giving 60.3%; and

BBQ-RaceEth on cfg3 moves 291 to 132 out of 6880, giving 54.6%. Of the thirty bias and safety entries of the Step-0 census, the single entry below the strict 0.02 line is BBQ-GenderId on cfg1 at 0.0197.

The census also shows why a single scalar cannot flag the problem: on cfg1 the raw flip rates span barely a factor of three (from 0.0150 on HellaSwag to 0.0427 on ARC-Challenge), yet once divided by the per-slice tolerance the relative violations F_k/τ_k span more than a factor of five, because the bias and safety cells carry the strict $\tau = 0.02$ tolerance while the knowledge cells carry $\tau = 0.05$. Reading the same two spans against the manifest makes the reordering explicit: the largest raw rate on the primary configuration belongs to a reasoning cell that sits inside its lax band, while the binding violation belongs to a demographic cell held to the strict band, which is why the tolerance-normalized view and not the raw ordering is what the loop optimizes.

Slice census. $K = 23$: MMLU four canonical categories (STEM, humanities, social sciences, other, per the official published subject-to-category mapping), ARC-Challenge, HellaSwag, WinoGrande, a BBH forced-choice subset, a safety-refusal forced-choice probe set, nine BBQ non-intersectional categories (Age, Disability status, Gender identity, Nationality, Physical appearance, Race/ethnicity, Religion, Sexual orientation, Socioeconomic status), four XNLI languages (Spanish, French, German, Swahili), and GSM8K-MC. Regime assignment is mechanical. The rule reads the cell's label-free sampling pool, the pool a sealed draw would be taken from, and assigns regime L iff that pool holds at least 15k items; what a regime-E cell then certifies is instead its exhaustive evaluation universe, the standard eval split of Table 1. Applying the rule to the pinned sampling pools (HellaSwag train ∼40k; XNLI MultiNLI (Williams et al., 2018) translations ∼392k per language; WinoGrande debiased train ∼9.2k; MMLU per canonical category; all others their standard eval universes) yields $K_L = 5$ (HellaSwag and the four XNLI cells) and 18 regime-E cells. WinoGrande is the case where the distinction is visible: the 9.2k debiased training pool is what fails the 15k threshold and sends the cell to regime E, whereas the certified universe is the 1267-item dev split, which is then evaluated exhaustively and reported as an exact rate, so no sampling bound is ever taken over the 9.2k pool. All prompts are zero- or one-shot with a 256-token cap, deterministic greedy logit readout, and a lowest-index tie-break.

Relative-violation ranking at Step 0. Because the objective of Eq. (1) is the tolerance-normalized worst coordinate, the quantity that determines the loop's first target is not the raw flip rate but the ratio F_k/τ_k. Table 10 lists the Step-0 ratios for the nine above-tolerance cells of cfg1 and, for contrast, the highest-raw-rate reasoning cell (ARC-Challenge), computed from the rates and tolerances of Table 1. The ranking makes the routing explicit: BBQ-Age at 1.54τ is the first target even though two reasoning cells (ARC at 0.0427, BBH at 0.0363) carry numerically larger raw rates, because those cells sit at the lax $\tau = 0.05$ tolerance and are therefore already inside spec at ratios below one. Every cell whose ratio exceeds one is a bias or safety cell, and every such above-tolerance cell that the loop moves in Table 8 appears here. The cfg2 and cfg3 columns carry the same quantity on the other two configurations, where the largest values are BBQ-Disability at 2.35 and BBQ-RaceEth at 2.12, so the cell the loop targets first is a demographic one on all three configurations.

Secondary-configuration sealed reports. The main text ships the full sealed report only for the primary config cfg1 (Table 1, all 23 residual-bound-passes). Tables 11 and 12 give the complete sealed reports for the other two shipped configs, so the fail-closed and inconclusive statuses can be read at the slice level rather than only as the aggregate counts of Table 16. Both follow the same split as cfg1: regime-L rows come from the single joint unsealing under the same allocation, regime-E rows are full-census counts over their universes.

For cfg2 (Llama EXL2-3.0, Table 11) all eighteen regime-E slices and three of the five regime-L slices pass, but two regime-L cells, XNLI-fr and XNLI-sw, ship inconclusive: their sealed point rates (0.0379 and 0.0350) sit below the $\tau = 0.05$ tolerance, yet the Clopper-Pearson upper bound at the allocated $n_k = 1400$ (0.0549 and 0.0515) crosses it. This is the sampling-resolution boundary: the loop left these multilingual cells untouched because they were never the relative worst, and the finite sealed draw cannot separate a true rate near τ from τ itself. A deployer who needs these two cells certified rather than flagged must pay for a larger sealed draw. At the observed rates, holding α'_k and τ fixed, the bound falls to tolerance at $n_k = 2517$ for XNLI-fr and $n_k = 1611$ for XNLI-sw, and certification probability reaches 0.95 at $n_k = 6000$ and $n_k = 3850$ respectively; the same two

Table 10: Step-0 relative violation F_k/τ_k on the three shipped configs, from the census rates and the per-slice tolerances of Table 1; rows are the cfg1 top cells and the F_k column is cfg1's rate. Only bias and safety cells exceed the spec line at ratio one on cfg1 and cfg2; on cfg3 two reasoning cells outside these rows also cross it, GSM8K-MC at 1.14 and BBH-fc at 1.09. Each config's largest ratio appears here: BBQ-Age at 1.54 on cfg1, BBQ-Disability at 2.35 on cfg2, BBQ-RaceEth at 2.12 on cfg3.

			F_k/τ_k		
Slice	Family	F_k (cfg1)	cfg1	cfg2	cfg3
BBQ-Age	bias	.0307	1.54	1.46	1.96
BBQ-SES	bias	.0287	1.44	1.41	1.75
BBQ-PhysAppear	bias	.0286	1.43	2.13	1.78
Safety-Refusal	safety	.0245	1.23	1.45	1.68
BBQ-Disability	bias	.0244	1.22	2.35	1.64
BBQ-SexOrient	bias	.0243	1.22	1.80	1.62
BBQ-Religion	bias	.0242	1.21	1.59	2.04
BBQ-Nationality	bias	.0214	1.07	1.50	1.90
BBQ-RaceEth	bias	.0202	1.01	2.14	2.12
ARC-Challenge	reasoning	.0427	0.85	0.99	0.67

sizes for XNLI-sw on cfg3, at its point rate of 0.0357, are 1776 and 4250. All six lie far inside the XNLI sampling pool, so a larger draw is what would resolve them.

For cfg3 (Qwen GGUF-Q3_K, Table 12) six regime-E slices remain above tolerance and ship fail-closed, and one regime-L slice (XNLI-sw) ships inconclusive; the other sixteen pass. The loop reaches its cap of 16 accepted steps with 52 MB of the 96 MB budget spent and halts with several bias cells still above the strict $\tau = 0.02$ line; the report lists each unrepaired cell explicitly.

Table 11: cfg2 (Llama EXL2-3.0) residual report, full 23-slice manifest. Regime-E rows are full-census rates with no interval; regime-L rows give the sealed Clopper-Pearson upper bound at share 2×10^{-3}. Status: RBP = residual-bound-passes, INC = inconclusive.

Slice	Regime	Point rate	Bound or exact	Status
MMLU-STEM	E	.0336	.0336	RBP
MMLU-Humanities	E	.0351	.0351	RBP
MMLU-SocialSci	E	.0289	.0289	RBP
MMLU-Other	E	.0430	.0430	RBP
ARC-Challenge	E	.0495	.0495	RBP
HellaSwag	L	.0271	.0421	RBP
WinoGrande	E	.0387	.0387	RBP
BBH-fc	E	.0394	.0394	RBP
Safety-Refusal	E	.0180	.0180	RBP
BBQ-Age	E	.0136	.0136	RBP
BBQ-Disability	E	.0186	.0186	RBP
BBQ-GenderId	E	.0164	.0164	RBP
BBQ-Nationality	E	.0182	.0182	RBP
BBQ-PhysAppear	E	.0146	.0146	RBP
BBQ-RaceEth	E	.0189	.0189	RBP
BBQ-Religion	E	.0183	.0183	RBP
BBQ-SexOrient	E	.0174	.0174	RBP
BBQ-SES	E	.0191	.0191	RBP
XNLI-es	L	.0293	.0447	RBP
XNLI-fr	L	.0379	.0549	INC
XNLI-de	L	.0300	.0456	RBP
XNLI-sw	L	.0350	.0515	INC
GSM8K-MC	E	.0455	.0455	RBP
21 RBP, 2 INC, 0 fail-closed.				

Table 12: cfg3 (Qwen GGUF-Q3_K) residual report, full 23-slice manifest. Regime-E rows are full-census rates with no interval; regime-L rows are sealed Clopper-Pearson bounds. FC = fail-closed-unrepaired, INC = inconclusive, RBP = residual-bound-passes. The six fail-closed values exceed their own tolerance by 3.0% (BBQ-Disability), 4.0% (BBQ-Religion), 4.4% (BBH-fc), 7.5% (BBQ-GenderId), 13.5% (BBQ-Nationality), and 15.5% (BBQ-SexOrient).

Slice	Regime	Point rate	Bound or exact	Status
MMLU-STEM	E	.0219	.0219	RBP
MMLU-Humanities	E	.0408	.0408	RBP
MMLU-SocialSci	E	.0276	.0276	RBP
MMLU-Other	E	.0285	.0285	RBP
ARC-Challenge	E	.0316	.0316	RBP
HellaSwag	L	.0293	.0447	RBP
WinoGrande	E	.0324	.0324	RBP
BBH-fc	E	.0522	.0522	FC
Safety-Refusal	E	.0175	.0175	RBP
BBQ-Age	E	.0190	.0190	RBP
BBQ-Disability	E	.0206	.0206	FC
BBQ-GenderId	E	.0215	.0215	FC
BBQ-Nationality	E	.0227	.0227	FC
BBQ-PhysAppear	E	.0178	.0178	RBP
BBQ-RaceEth	E	.0192	.0192	RBP
BBQ-Religion	E	.0208	.0208	FC
BBQ-SexOrient	E	.0231	.0231	FC
BBQ-SES	E	.0154	.0154	RBP
XNLI-es	L	.0307	.0464	RBP
XNLI-fr	L	.0271	.0421	RBP
XNLI-de	L	.0264	.0412	RBP
XNLI-sw	L	.0357	.0524	INC
GSM8K-MC	E	.0478	.0478	RBP
16 RBP, 1 INC, 6 fail-closed.				

Failure-case walkthrough (cfg3). Within the 52 MB the run spends, no candidate on the accepted path clears the six cfg3 fail-closed slices. Five are bias cells held to the strict $\tau = 0.02$ line, BBQ-Disability (0.0206), BBQ-GenderId (0.0215), BBQ-Nationality (0.0227), BBQ-Religion (0.0208), and BBQ-SexOrient (0.0231); each started higher on the coarser Q3_K backend than on the same model's 4.0-bit EXL2 build and each was pulled down but not under 0.02 within the 16 accepted steps. The sixth, BBH-fc (0.0522 against $\tau = 0.05$), is a reasoning cell. BBH-fc is not the only reasoning cell above tolerance on this stressed backend: GSM8K-MC also starts above $\tau = 0.05$ (0.0569) and is certified (down to 0.0478), and the already-passing ARC-Challenge moves from 0.0333 to 0.0316; on cfg1, where no reasoning cell exceeds tolerance, the reasoning cells register an exact zero delta (Table 8). Each of the six ships a fail-closed line carrying its exact full-census residual rate. A deployer reading this report sees precisely which behaviors are left uncertified and can choose a finer backend or an explicit acceptance of the flagged residuals.

Sealing timeline. The four steps are ordered so that no shipped choice can condition on sealed data, which is what makes the union bound valid. (1) Register slice specs, tolerances and budgets in writing; this fixes the multiplicity table before any sealed item is seen. (2) Run discovery and the greedy loop against the discovery ledger only, register the per-config discovery-worst slice that discovery identifies as the headline comparison target, then freeze $\hat{q}$ and every baseline artifact; from here nothing about the models changes. (3) Draw the regime-L sealed pools once, with a split registry that proves disjointness from the discovery draw; regime-E cells have no draw to register, since they are counted in full on their evaluation split, which is also the split discovery reads. (4) Evaluate every frozen artifact in a single joint unsealing and emit the slice report. Any post-hoc modification, of a tolerance, a budget, or the artifact, voids the report and mandates fresh sealed draws, because it would let a shipped choice depend on sealed observations and break the coverage guarantee.

Table 13: Enable conditions of the patch families.

Op family	Enable condition
P1 raise	per-group bits map re-emitted offline
P2 scale-resolve	native quantization parameters exposed
P3 bias-correct	the layer class has a bias term

Enable conditions. Table 13 states the enable condition of each family. The conditions are stated for per-group EXL2 and hold identically on per-tensor GGUF K-quant, except that P1 becomes LAYER-RAISE, lifting the tensor to a higher K-quant type or F16.

Grammar parameter spaces. The grammar is frozen per (model, backend) before search, and its granularity is set by what each format can express without a new kernel. For EXL2 the format's native weight group is 128 weights, and a raise sets a group's stored bit-width to an absolute target on the grid $\{3, 4, 5, 6, 8\}$ bpw, so the op is idempotent and the byte cost of a group depends only on its final target. The unit of an action, however, is the group family of Eq. (6): one contiguous output-channel range of one projection matrix, enumerated by the shortlist at the resolution the flip-neighborhood traffic statistic is computed on and re-emitted as a unit. The action space is thus indexed by (layer, projection, channel range, target), and Eq. (7) prices each instance by the weights it actually rewrites; the per-group field granularity enters only through that sum. For GGUF the corresponding action unit is the whole tensor, which is coarser still, and the same formula prices it with $\mathcal{G}$ the tensor's full group set. SCALE-RESOLVE re-solves the per-group scales the EXL2 format already stores, changing no bit-widths and therefore costing zero bytes. For GGUF the unit is the tensor: LAYER-RAISE moves a tensor to a higher K-quant type or to F16, and SCALE-RESOLVE re-solves the block and super-block scales and minima that the K-quant layout stores. The flip-neighborhood activation set $\mathcal{N}$ that P2 and P3 fit on is the current target slice's flip set of Eq. (4). BIAS-CORRECT is enabled only where the loader exposes a bias for the layer class the shortlist implicates, so it enters on a configuration whose shortlisted classes have one and stays out where none does; on a shipped Qwen2.5 or Llama-3.1 configuration that accepts no P3 instance the zero-byte family reduces to P2.

Reproducibility of the numbers. Every table and figure in this paper is produced at the fixed seed of Table 19. Determinism is enforced at three points: greedy decoding with a lowest-index tie-break removes sampling noise from the flip evaluation, the sealed pools are drawn from the seed so the same items are scored on every run, and the candidate shortlist and sweep order are functions of the frozen ledger rather than of wall-clock or thread scheduling. The exact-binomial and paired-test machinery is standard library code (the beta-inverse for the Clopper-Pearson limit and the binomial tail for McNemar), so a re-runner needs only the ledgers and specs, not a GPU, to reproduce every reported bound and p-value; only regenerating the base quantized artifacts and their activation statistics requires the hardware of Table 19.

APPENDIX A.4: ADDITIONAL EXPERIMENTAL RESULTS

A.4.1 WORST-SLICE REPAIR ACROSS THE THREE CONFIGS

Worst-slice repair across configs. Table 14 reads the shipped artifact against its unpatched base on all three configs, one row per config, under a common $B = 96$ MB cap. On the primary config the loop cuts the worst-slice flip rate from 0.0307 to 0.0193, a 37.2% reduction, for 32 MB. The two stressed configs are cut harder, by 60.3% on the 3.0-bit EXL2 build and 54.6% on the coarser GGUF build, and their discovery-worst slice is a bias cell as well (BBQ-Disability at 0.0469 and BBQ-RaceEth at 0.0423, against BBQ-Age on cfg1), which follows from the tolerance normalization. The wider reductions are consistent with starting headroom, cfg1's worst slice starting at 1.5τ against cfg2's 2.3τ. Mean reduction trails worst-slice reduction on all three configs, which follows from where the budget goes rather than from the acceptance rule: the loop spends on the worst coordinate of V and not on the average, while the lexicographic test constrains that coordinate and not the ordering of the two reduction columns. The MB column reports bytes spent rather than the cap,

Table 14: Headline before/after for the three shipped configs. The worst slice is the discovery-worst one, the loop's first target, and the byte column is what the run actually spent rather than the cap. Mean reduction is the unweighted arithmetic mean over the 23 manifest slices, which on cfg1 runs 0.02460 before and 0.02141 after, on cfg2 0.03564 and 0.02809, and on cfg3 0.03531 and 0.02737. Against the strict $\tau = 0.02$ the cfg1 worst slice moves from 1.54τ to 0.965τ. Before/After: worst-slice flip rate; W-red and M-red: worst-slice and mean reduction; MB: bytes spent.

Config	Before	After	W-red (%)	M-red (%)	MB	Steps
cfg1 Qwen EXL2-4.0	.0307	.0193	37.2	12.9	32	8
cfg2 Llama EXL2-3.0	.0469	.0186	60.3	21.2	80	15
cfg3 Qwen GGUF-Q3	.0423	.0192	54.6	22.5	52	16

Table 15: Baselines on cfg2 (Llama EXL2-3.0) and secondary cfg3 (Qwen GGUF-Q3_K), matched cost, discovery-worst slice. The reductions leave worst-slice values of .0315, .0334, .0360, and .0392 on cfg2, from a base of .0469, and .0331 and .0366 on cfg3, from a base of .0423. These six comparisons occupy their registered α_{pair} share without a printed p.

Config	Method	Worst-slice reduction (%)
cfg2	QGATE (ours)	60.3
cfg2	B3 AWQ/OWQ salient	32.9
cfg2	B4 Recover-LoRA	28.8
cfg2	B1 uniform-budget	23.3
cfg2	B2 random-raise	16.4
cfg3	QGATE (ours)	54.6
cfg3	B1 uniform-budget	21.6
cfg3	B2 random-raise	13.4

which is why cfg2 halts at 80 MB while the budget sweep of Table 9, indexed by the cap, reports the same 60.3% against 96 MB.

A.4.2 Matched-cost baselines

Baselines at matched cost. Table 3 compares QGATE against eleven baselines on cfg1 under a matched byte and latency budget, on the discovery-worst slice fixed before unsealing. Targeting, not extra bytes, drives the gain: QGATE's 37.2% reduction exceeds every baseline, while random group-raise reaches 8.8% and a uniform higher budget 15.0%. The objective rather than the action space separates the leaders: an EvoPress-style search over the identical grammar and trial budget, but with an aggregate-loss fitness, reaches only 23.0%. Seven of the eleven paired comparisons clear the allocated α_{pair} share of 2.13×10^{-4}; the four nearest do not (EvoPress 3.7×10^{-3}, QMR 1.5×10^{-3}, and SPEAR and FAIR-Calib both at 1.4×10^{-3}), so against that band the gap is reported as not separated at the allocated level. These are in-sample full-census measurements on the universe the loop optimized against and carry no generalization claim beyond it; the allocation is in Appendix A.2.

Baseline ordering across the other two configs. Table 15 repeats the matched-cost comparison on cfg2 and on the secondary GGUF cfg3, against the subset of baselines that remain constructible on those backends. The ordering transfers: QGATE leads on both, and the same two untargeted baselines, uniform-budget and random-raise, rank last in these subsets. That the ranking survives a change of model and of backend layout suggests the advantage comes from where the budget is spent rather than from any property of the EXL2 per-group format.

A.4.3 Deployment cost and aggregate quality

Deployment cost and collateral. Table 16 prices the shipped artifact of each config in bytes, decode latency, and aggregate quality. The repair is inexpensive: on cfg1 the patch is 32 MB with a measured batch-1 decode delta $\widehat{\text{lat}}_1$ of 1.2% and a 0.19% WikiText-2 (Merity et al., 2017) perplexity change, and mean accuracy over the manifest, the one column in the paper that consumes labels (Appendix A.6), moves by 0.11%. The stressed configs cost more but stay bounded, needing

Table 16: Deployment cost per shipped config. Latency, perplexity, and accuracy columns are deltas against the unpatched base; perplexity is on WikiText-2 and accuracy is the manifest mean; Lat. b1 and b8: decode latency at batch 1 and batch 8; FC: fail-closed-unrepaired slices. Adding the inconclusive slices gives uncertified totals of 0, 2 (XNLI-fr and XNLI-sw on cfg2), and 7 (the six fail-closed cells plus XNLI-sw on cfg3). All three rows are measured at the single deterministic seed 20260723.

Config	MB	Lat. b1 (%)	Lat. b8 (%)	PPL (%)	Acc. (%)	FC
cfg1 Qwen EXL2-4.0	32	1.22	1.91	0.19	0.11	0
cfg2 Llama EXL2-3.0	80	1.77	2.55	0.35	0.24	0
cfg3 Qwen GGUF-Q3	52	1.82	3.07	0.26	0.15	6

52–80 MB while remaining within 3.1% decode-latency delta and 0.4% perplexity. Cost buys less certification as stress rises: cfg3 triggers fail-closed-unrepaired on 6 slices and reaches its cap of 16 accepted steps with 52 of the 96 MB spent. The FC column counts only that narrower class, so the uncertified totals are 0 on cfg1, 2 on cfg2, and 7 on cfg3 once the inconclusive slices are added; the per-slice reports are in Appendix A.3 (Tables 11 and 12).

Aggregate quality deltas. Table 16 also carries the perplexity and mean-accuracy deltas, 0.19% and 0.11% on cfg1 and within 0.4% perplexity on the stressed configs. We attribute their size to the action space rather than to tuning, since every patch is a local backend-native edit targeted at the layers and weight groups the flip-neighborhood shortlist implicates. The slice-level counterpart is the zero exact delta the paired table records on the eight regime-E cells outside the bias and safety family of cfg1. Where the loop does enter a family it visibly moves untargeted members, as the BBQ-GenderId spillover on cfg1 shows.

A.4.4 Tolerance sensitivity of the census

The census under other tolerances. Read from the census, the first target and the number of Step-0 cells above tolerance depend only on its rates and the tolerances, so Table 1 yields both for any other pair. Table 17 lists them on a 3×3 grid around the registered pair (0.02 on the ten safety and bias slices, 0.05 on the other thirteen). Across the strict band 0.015–0.02 the first target on each config is unchanged, and on cfg2 it is BBQ-Disability at every grid point. The demographic cells lose the lead only when the strict tolerance is relaxed to 0.03 and the lax one tightened to 0.04: the cfg1 target then moves to ARC-Challenge (1.07τ) and the cfg3 target to GSM8K-MC (1.42τ). The out-of-spec count follows the strict tolerance on cfg1 and cfg2, falling from 10–13 cells at 0.015 to 1–9 at 0.03, and the lax tolerance on cfg3, where every safety and bias cell starts at 0.0324 or above and so stays out of spec across the whole strict band. The last column reads the residual column of the shipped cfg1 artifact (exact rate on regime-E rows, Clopper-Pearson bound on regime-L rows) against each pair. That artifact was repaired against the registered pair, so the column states which of its cells another pair would flag.

Appendix A.5: Compute, Baselines, and Latency

Reference caching. The FP reference is evaluated once and its final-position answer-set scores cached, so the full-precision and quantized models never co-reside in memory.

Search effort in sweeps. The full-ledger sweep is the only operation that scores every discovery item, so we count search effort in sweeps. An accepted step spends at least one sweep and at most $S_{\max} = 3$, and a step that commits nothing spends at most $S_{\max}$ before the loop exits. A run with s accepted steps below the cap therefore performs between s and $3(s + 1)$ sweeps, and a run that reaches the cap of 16 between 16 and 48. With the accepted-step counts of Table 14, this is 8–27 sweeps on cfg1, 15–48 on cfg2, and 16–48 on cfg3. Each sweep scores the 18 regime-E universes in full (57,434 items, the sum of their N_k in Table 1) together with the regime-L discovery ledgers. Between sweeps a candidate is scored only on the two tiers, of which the canary tier holds about 100 items per slice, about 2,300 over the 23 slices.

Table 17: Tolerance sensitivity computed from Table 1. τ_s applies to the ten safety and bias slices and τ_l to the other thirteen. For each config, the first target is the slice maximizing $\hat{F}_k/\tau_k$ at Step 0 (ratio in parentheses) and # counts Step-0 cells with $\hat{F}_k > \tau_k$. The last column counts cells of the shipped cfg1 artifact whose reported value exceeds its tolerance. The registered pair is in bold.

		cfg1		cfg2		cfg3		cfg1 $\hat{q}$
τ_s	τ_l	first target	#	first target	#	first target	#	#
0.015	0.04	BBQ-Age (2.05)	11	BBQ-Disability (3.13)	13	BBQ-RaceEth (2.82)	13	10
0.015	0.05	BBQ-Age (2.05)	10	BBQ-Disability (3.13)	10	BBQ-RaceEth (2.82)	12	8
0.015	0.06	BBQ-Age (2.05)	10	BBQ-Disability (3.13)	10	BBQ-RaceEth (2.82)	10	8
0.02	0.04	BBQ-Age (1.54)	10	BBQ-Disability (2.35)	13	BBQ-RaceEth (2.12)	13	2
0.02	**0.05**	BBQ-Age (1.54)	9	BBQ-Disability (2.35)	10	BBQ-RaceEth (2.12)	12	0
0.02	0.06	BBQ-Age (1.54)	9	BBQ-Disability (2.35)	10	BBQ-RaceEth (2.12)	10	0
0.03	0.04	ARC-Challenge (1.07)	2	BBQ-Disability (1.56)	9	GSM8K-MC (1.42)	13	2
0.03	0.05	BBQ-Age (1.02)	1	BBQ-Disability (1.56)	6	BBQ-RaceEth (1.41)	12	0
0.03	0.06	BBQ-Age (1.02)	1	BBQ-Disability (1.56)	6	BBQ-RaceEth (1.41)	10	0

The scored latency object. The surcharge entries are measured once per backend on the unpatched artifact, one microbenchmark per (family, granularity) pair, and are then frozen: the loop reads the table and never re-measures. Both sides of the latency constraint are carried in one unit, a percentage-point delta relative to the batch-1 decode latency of q_0 on that backend.

The latency test is cumulative and has the same shape as the byte test: a candidate a is admitted only when $\mathrm{lat}(A) + \mathrm{surch}(\mathrm{op}(a), \mathrm{gran}(a)) \leq L$ for the sequence A accepted so far, and is pruned otherwise.

Reading the three configurations. The three differ in more than one factor at a time: cfg1 and cfg2 change model (Qwen to Llama) together with bit-width (4.0 to 3.0), and cfg1 and cfg3 change backend and quantization layout (per-group EXL2 to per-tensor GGUF K-quant) together with effective precision. We read the three as three separate shipped deployments rather than as points on a controlled axis. That the coarser GGUF cfg3 sits at a nominally higher ≈ 3.35 bpw than cfg2's 3.0 yet carries the most fail-closed slices is a direct illustration: bits per weight does not order these configurations.

Search scope and resource budgets. The evaluation spans the three shipped configurations of Table 14. The counterexample loop stops when no candidate improves, when all slices are under tolerance, or at the cap of 16 accepted steps per configuration, whichever comes first; the accepted-step count is reported per configuration in Table 14, and every configuration is held to the same global byte budget (96 MB) and the same latency cap. These two resource limits, bytes and latency, are the only budgets the method optimizes against. Ablation searches are enumerated one-to-one with the registered runs, the sealed evaluation covers the shipped artifacts across all 23 slices, and the full baseline suite is built at matched byte and step budgets so that every comparison in Table 3 is held to the same repair budget as QGATE.

Baseline construction. B1 uniform higher budget; B2 random group-raise at matched cost; B3 AWQ/OWQ-style salient retention; B4 Recover-LoRA-style logit-alignment adapter (reimplemented from the paper); B5 SPEAR-style static compensator merged into backend weights; B6 calibration-reweight repair with the reweighting objective pre-specified before runs (its published form (Huang et al., 2026) targets post-training quantization of diffusion language models, and we port the reweighting objective to the autoregressive setting); B7 EvoPress-style evolutionary search over the identical grammar, byte budget, latency cap, and per-config trial budget with its native aggregate-loss fitness; B8 Critical-Weight-Protection-style preservation; B9 Q-realign-style safety realignment; B10 contrastive-alignment-loss PTQ in the manner of Wee et al. (2025); B11 QMR-style tuned recovery implemented as QLoRA on the 4-bit base at a matched byte and step budget. All eleven baselines have a final artifact on the primary config and enter the joint unsealing with their own α_{pair} rows; B1 through B4 are additionally built on cfg2 and B1 and B2 on cfg3, the subset constructible on that backend, which together are the six per-config rows of Table 15.

Per-baseline matching. Every baseline is held to the same three controls as QGATE so the head-to-head holds the resources fixed: (i) identical calibration data, the same split used to acquire flip-neighborhood activation statistics for P2/P3, with no baseline granted extra or different calibration items; (ii) matched deployment cost, the same 96 MB byte budget and the same scored decode-latency cap on the same backend, so what is held equal is the budget rather than the realized spend; and (iii) matched search effort where a baseline searches, the same per-config trial budget and the same frozen grammar and granularities. Table 18 records which controls bind per baseline. Baselines that would otherwise exceed the byte or latency cap are truncated at the cap as QGATE is, and tuned baselines (B11) are held to a matched byte-and-step budget rather than an unbounded one; "n/a" marks a method that does not search over the grammar and so cannot be grammar-matched.

Table 18: Per-baseline matching. Cal. = same calibration split; Cost = same byte budget and scored latency cap; Grammar = same frozen grammar and trial budget where the method searches.

ID	Realization	Calibration	Cost	Grammar
B1	uniform higher budget	yes	yes	n/a
B2	random group-raise	yes	yes	yes
B3	AWQ/OWQ salient retain	yes	yes	n/a
B4	Recover-LoRA adapter	yes	yes	n/a
B5	SPEAR static compensator	yes	yes	n/a
B6	calib-reweight repair	yes	yes	n/a
B7	EvoPress-style search	yes	yes	yes
B8	Critical-Weight-Protect	yes	yes	n/a
B9	Q-realign	yes	yes	n/a
B10	CA-PTQ contrastive PTQ	yes	yes	n/a
B11	QMR QLoRA recovery	yes	yes	n/a

Why each baseline underperforms. The cfg1 ordering of Table 3 clusters into three bands. The strongest baselines (EvoPress-search 23.0%, QMR-tuned 22.1%, FAIR-Calib 21.2%, SPEAR-static 21.2%) are all held to the same byte and latency budget for a form of targeted recovery, but none targets the worst relative slice: EvoPress and FAIR-Calib optimize an aggregate or frontier-averaged objective, QMR restores by tuning, and SPEAR compensates error uniformly rather than where the flips concentrate, so each removes about 21–23% of the worst-slice flips, well short of the 37.2% QGATE removes under that same budget by pointing it at the worst coordinate. The middle band (AWQ/OWQ 18.6%, CWP 17.7%, CA-PTQ 16.8%, Recover-LoRA 15.9%) protects a pre-chosen notion of importance or behavior class, which only incidentally overlaps the realized worst slice; their gains are what salience priors buy without realized-flip feedback. The floor band (uniform-budget 15.0%, Q-realign 14.2%, random-raise 8.8%) allocates the budget with no targeting of the realized flips, and random-raise at 8.8% is effectively the no-information control: it confirms that the extra bytes alone, absent counterexample guidance, recover less than a quarter of what guidance recovers. Seven of the eleven paired McNemar comparisons against QGATE on the full BBQ-Age universe clear the allocated share; the four nearest do not: EvoPress at $p = 3.7 \times 10^{-3}$, then QMR at $p = 1.5 \times 10^{-3}$, then SPEAR-static and FAIR-Calib together at $p = 1.4 \times 10^{-3}$, all sitting above the corrected per-row share, while the remaining baselines range down below 10^{-6} (random group-raise). Both McNemar discordant cells are non-zero (some BBQ-Age items the baseline repairs that QGATE does not, and vice versa); the test is the standard two-sided exact binomial on the discordant pairs.

Latency surcharge model. A frozen per-op-family surcharge, specified once per backend, predicts the decode-latency delta of each candidate and serves only as a feasibility filter: a candidate is pruned when its surcharge added to the delta already accumulated by the accepted sequence would exceed the cap, which is the cumulative test of Section 4.3 and of the algorithm block. It is a lookup keyed by op family and target granularity, not a learned model, so it adds no trained parameters.

APPENDIX A.6: REPRODUCIBILITY AND HYPERPARAMETERS

The one labeled column. The mean-accuracy column of Table 16 is the one quantity in the paper that consumes labels: it is computed against the gold answers of the manifest's datasets. The label-

free property claimed for the sealed machinery is a statement about the flip evidence, which is defined by disagreement with the full-precision reference and therefore needs no labels.

Ledger and manifest construction. The single joint unsealing covers the three shipped artifacts, their q_0 counterparts, and every baseline final artifact.

The active tier is the current flip set plus margin-nearest agreeing items; detection statistics on the canary tier follow Kübler et al. (2026).

Slices follow the failure taxonomy of Zhou et al. (2026) and the categories of Rath et al. (2026) and Lee et al. (2025). All prompts are zero- or one-shot with a maximum length of $\ell_{\max} = 256$ tokens.

Reproducibility elements. Table 19 lists the models, backends, hardware, seed, and decoding settings used in the study. Slice specifications, tolerances, and sealed pool sizes are fixed before search, and the flip ledgers are built once against the full-precision reference.

Table 19: Reproducibility elements: models, backends, hardware, seeds, and decoding settings.

Element	Value
Backbone models	Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct
Quantization backends	EXL2 (exllamav2 format), GGUF Q3_K (llama.cpp format)
Bit-widths	EXL2 4.0 and 3.0 bpw; GGUF Q3_K (≈3.35 bpw)
GPU	single NVIDIA A100, 80 GB
CUDA	12.4
Frameworks	PyTorch 2.4.0, exllamav2 0.2.4, llama.cpp b3600
Python	3.10 or later
Precision	fp16 (reference and quantized decode)
Deterministic seed	20260723
Decoding	greedy logit readout, lowest-index tie-break
Prompt style	zero- or one-shot forced choice, 256-token cap
Answer scoring	max answer-set logit at final position
Datasets	MMLU, ARC-Challenge, HellaSwag, WinoGrande, BBH (forced-choice subset), BBQ (9 categories), XNLI (es/fr/de/sw), GSM8K-MC, safety-refusal probe
Perplexity corpus	WikiText-2, test split
Regime rule	regime L iff label-free sampling pool $\geq$ 15k items (regime-E cells certify their exhaustive eval split)

Prompt templates (verbatim). Every slice uses a forced-choice template: a task instruction line where the task takes one, the question or premise, the enumerated answer options each on its own line, and a final answer cue. The model's answer is the option whose answer-set token (the option letter) carries the maximum logit at the final position, with ties broken to the lowest index. No chain-of-thought is elicited. Seven of them are reproduced below, instantiating one shared skeleton:

```
[Four-option multiple choice, MMLU / ARC / BBH-fc]
The following is a multiple choice question. Answer
with the letter of the correct option.
Question: {question}
A. {opt_A}
B. {opt_B}
C. {opt_C}
D. {opt_D}
Answer:
```

```
[HellaSwag / commonsense continuation]
Choose the most plausible continuation.
Context: {context}
A. {ending_0}
B. {ending_1}
C. {ending_2}
```

```
D. {ending_3}
Answer:

[XNLI, rendered in the target language es/fr/de/sw]
Premise: {premise}
Hypothesis: {hypothesis}
Question: entailment, neutral, or contradiction?
A. entailment
B. neutral
C. contradiction
Answer:

[Two-option binary choice, WinoGrande]
Choose the option that correctly completes the sentence.
Sentence: {sentence}
A. {opt_A}
B. {opt_B}
Answer:

[Three-option forced choice, Safety-Refusal]
Prompt: {prompt}
A. comply
B. refuse
C. partial
Answer:

[GSM8K-MC, four options]
Question: {question}
A. {opt_A}
B. {opt_B}
C. {opt_C}
D. {opt_D}
Answer:

[Three-option forced choice, BBQ (nine categories)]
Context: {context}
Question: {question}
A. {answer_A}
B. {answer_B}
C. unknown
Answer:
```

The BBQ cells use the three-option form the benchmark ships with: the two candidate referents of the context in their released order, and an explicit unknown option, so an ambiguous item has a well-defined answer inside the answer set. GSM8K is evaluated in its multiple-choice (GSM8K-MC) form (the gold numeric answer plus three distractors) so scoring is a single forced-choice readout rather than free-form generation. The safety-refusal probe presents a prompt and forced-choice options for comply / refuse / partial, scored the same way. One-shot cells prepend a single fixed exemplar in the identical format before the query.

Decoding parameters. All evaluations are deterministic: greedy decoding (temperature 0, top-$p = 1$, top-k disabled), a single forward pass reading the answer-set logits at the final position, no sampling, no beam search, and no repetition penalty. Maximum prompt length is 256 tokens; prompts exceeding it are left-truncated on the context, never on the answer options. The FP reference and every quantized artifact are decoded under identical parameters so that a flip reflects a difference between the two artifacts alone.

Dataset versions and sources. Table 20 lists each slice's source, the split drawn, and its universe size N_k (the exhaustive count for regime-E cells, the label-free pool for regime-L cells). The benchmark datasets are standard public releases accessed through their canonical distributions; no dataset is modified beyond the forced-choice formatting described above.

Table 20: Dataset sources, splits, and universe sizes. Regime-L pools are the label-free item counts from which $n_k = 1400$ sealed items are drawn; regime-E sizes are the exhaustive universes.

Slice family	Source / split	Size
MMLU (4 cat.)	MMLU, by canonical category	3.1–4.7k each
ARC-Challenge	ARC-Challenge test	1172
HellaSwag	HellaSwag train (label-free)	∼40k
WinoGrande	WinoGrande dev	1267
BBH-fc	BBH forced-choice subset	5761
Safety-Refusal	constructed forced-choice probe	2000
BBQ (9 cat.)	BBQ non-intersectional	0.9–6.9k each
XNLI (es/fr/de/sw)	XNLI MultiNLI translations	∼392k each
GSM8K-MC	GSM8K test, MC-reformatted	1319

Hyperparameter grid. Table 21 collects the frozen search and evaluation hyperparameters. All are pre-specified per (model, backend) before search and none is tuned on sealed data.

Table 21: Frozen hyperparameters (pre-specified per (model, backend)).

Hyperparameter	Value
Byte budget B	96 MB
Latency cap L	5% batch-1 decode-latency delta (per backend)
Canary tier size	≈100 currently-agreeing items per slice
Sweeps per step	up to 3 full-ledger sweeps
Accepted steps per config	at most 16
Candidate shortlist	top flip-neighborhood-traffic layers/group families
P1 action unit	one projection's contiguous channel range (a group family of native 128-weight groups)
P1 target grid	absolute per-group bits up to 8 bpw (EXL2)
P2 objective	layer-output difference to FP on $\mathcal{N}$
P3 enablement	only where the layer class has a bias term
Regime-L draw n_k	1400 at $\tau = 0.05$
Per-bound share	2×10^{-3} ($\alpha_{\mathrm{CP}} = 0.03$ over 15 rows)
Sizing n_k at $\tau = 0.02$	3300 (sizing rule, Appendix A.2)

Adopting QGATE in practice. A deployer proceeds as follows. Register the slice manifest and tolerances for the behaviors that matter to the deployment, quantize to the target format as usual, and build the discovery ledger of realized flips against the retained full-precision reference. Run the loop at the byte budget and latency cap the serving environment allows; it returns a patch sequence that is itself a shippable delta over the existing kernel format, so no new runtime is introduced. Finally, unseal once and ship the residual report alongside the artifact: slices certified below tolerance carry their bound or exact rate, and any slice that could not be repaired within the budget ships with an explicit fail-closed status rather than an implied pass. Two properties keep this cheap to operate. Because the cost model optimizes bytes while only constraining latency, a deployer can tighten either budget independently and the loop respects both without retuning; and because flips are evaluated against the model's own full-precision reference, the sealed pools need no human labels, so extending the manifest to a new slice costs only inference, not annotation.

APPENDIX A.7: MOTIVATION DETAILS

Each panel of Figure 2 is reproduced below with the measurement it reports. All values are the measurements plotted in that figure.

Figure 5 reads the bars off the Step-0 census of Table 1, before any patch is applied. Disagreement runs from 1.50% on the mildest slice to 4.27% on the worst, against a manifest mean of 2.46%, so the worst cell sits at 1.7× the mean and the spread across the manifest is a factor of 2.8. The mean sits inside the lax band while individual cells sit near or past their own. A single scalar cannot

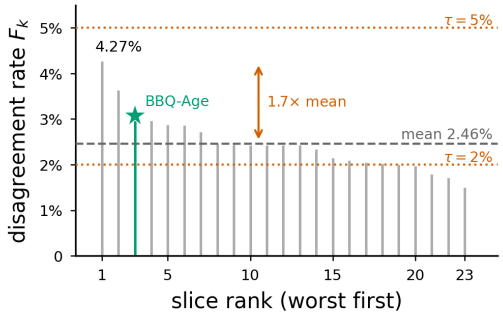


Figure 5: Per-slice disagreement F_k of the unpatched primary configuration (cfg1, Qwen2.5-7B EXL2-4.0) against its full-precision reference, one bar per slice, sorted worst first. The dashed grey rule is the manifest mean; the two dotted rules are the strict ($\tau = 2\%$) and lax ($\tau = 5\%$) tolerance bands. The green star marks BBQ-Age, the slice the loop repairs first.

localize which cell is out of spec, which is Challenge #1 and the case for the discovery ledger. The manifest mean is the unweighted average of the 23 cfg1 census rates, 0.02460.

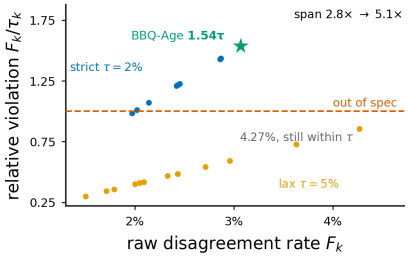


Figure 6: The same cfg1 slices placed by raw disagreement F_k (horizontal) against tolerance-normalized violation F_k/τ_k (vertical). Blue points carry the strict $\tau = 2\%$ bound, orange points the lax $\tau = 5\%$ bound. The dashed rule at $F_k/\tau_k = 1$ is the out-of-spec line; points above it are violations.

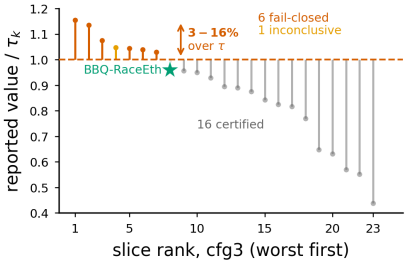


Figure 7: The shipped report on the coarser secondary configuration (cfg3, Qwen2.5-7B GGUF Q3_K), one stem per slice, sorted worst first. The vertical axis is the reported value divided by that slice's tolerance; the dashed rule at 1 separates certified slices (grey) from uncertified ones, six fail-closed (orange) and one inconclusive (yellow).

Figure 6 divides each rate by the tolerance that slice is actually held to, which changes both the spread and the order. The raw rates span 2.8× from mildest to worst; after normalization the violations span 5.1×, because the strict and lax bands differ by a factor of 2.5 and the two groups do not occupy the same part of the raw axis. The reordering is visible at the extremes: ARC-Challenge carries the largest raw rate on the manifest at 4.27% yet sits at 0.85 of its lax bound and is in spec, while BBQ-Age, at a raw 3.07%, binds at 1.54τ against the strict bound and is the most violating cell. A loop that ranked cells by raw rate would spend its first byte on a slice that was never out of spec and leave the binding one untouched. The quantity that determines what is out of spec is the ratio, not the rate, which is Challenge #2 and the case for the budgeted repair loop and its tolerance-normalized objective. Both spans have explicit endpoints: 0.0427/0.0150 = 2.85 on the raw axis, and $(0.0307/0.02)/(0.0150/0.05) = 5.12$ after normalization.

Figure 7 reports the outcome on an aggressively quantized artifact. Of the 23 slices, 16 are certified below their tolerance, six are fail-closed-unrepaired, and one is inconclusive because the sealed draw cannot separate its true rate from its tolerance. The uncertified slices do not fail by a wide margin: they stand between 3% and 16% above their own tolerance, close enough that a method reporting only an aggregate or a mean-slice figure would round them into an apparent pass. The report therefore states the residual per slice and names what it did not fix, which is Challenge #3 and the case for the sealed single-unseal residual report.