PREDICTION INSTABILITY AND TASK UTILITY IN LOW-PRECISION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Selective precision and quantization repair are evaluated through changes in predictions, reconstruction error, and task accuracy. We examine how these measurements relate using a controlled audit of low-precision language models. A prespecified experiment evaluates two instruction-tuned models on 1,024 HellaSwag training examples across four precision paths and three execution layouts. Changing option batch size while preserving padding changes per-item precision-upgrade utility on 42 examples for Qwen and 14 for SmolLM2; corresponding full-model FP32 arithmetic controls show zero changes. Within fixed 256-example margin selections, opposing changes cancel, leaving aggregate shifts of -6 and $+1$ net-correct units. Subsequent SmolLM2 repair diagnostics compare four teacher-target objectives with the same rank-16 parameterization. Centered-score fitting reduces training error by 46.17%, while error on two evaluation panels increases by 15.24% and 8.16%. On the previously observed 1,024-question transfer panel, it yields three additional correct answers, compared with five for option-level distillation. An ARC replay further distinguishes offline fallback utility from executed predictions and cost. The measurements identify where prediction stability and reconstruction fidelity diverge from task utility, and provide a protocol for evaluating these quantities together.

1 INTRODUCTION

An uncertainty score can identify inputs on which a low-precision model is likely to be wrong without identifying inputs on which high-precision recomputation will help. A precision upgrade can repair an answer, leave its correctness unchanged, or replace a correct answer with an incorrect one. Selective recomputation therefore depends on the *difference* between two outcomes. When execution layout also changes those outcomes, an offline estimate of upgrade utility is conditional on more than the selected model weights and examples.

Prior work documents offsetting correctness flips under quantization despite similar aggregate accuracy (Dutta et al., 2024). Batch size and other system configurations can alter low-precision inference, while FP32 arithmetic can improve stability (Yuan et al., 2025). Margin-triggered verification addresses batch-dependent token decisions (Chu et al., 2026), and paired-margin models predict quantization-induced decisions under calibrated conditions (Wu et al., 2026). Reconstruction methods address a complementary problem: recovering layer outputs from quantized weights through low-rank corrections (Zhang et al., 2024; 2025). Prediction stability, reconstruction error, and task correctness measure different properties of these interventions.

Our primary question is whether changes in per-example precision-upgrade utility produce a comparably large change in the utility of a fixed fallback policy. A policy selects a particular set of inputs, whereas a whole-panel instability count omits both membership in that set and the direction of each change. We hold the selected set fixed across layouts and distinguish exact repeat consistency, item-level utility drift, and aggregate policy drift. A subsequent repair study asks whether improving a teacher-matching objective transfers to answer correctness under a fixed correction architecture.

The study has three empirical components. The prespecified two-model experiment connects item-level transitions to fixed-policy utility using native arithmetic and effective-weight-matched FP32 controls. An execution replay compares offline estimates with realized fallback outcomes and cost. Teacher-target repair experiments compare hidden-state reconstruction, centered-score fitting, and

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

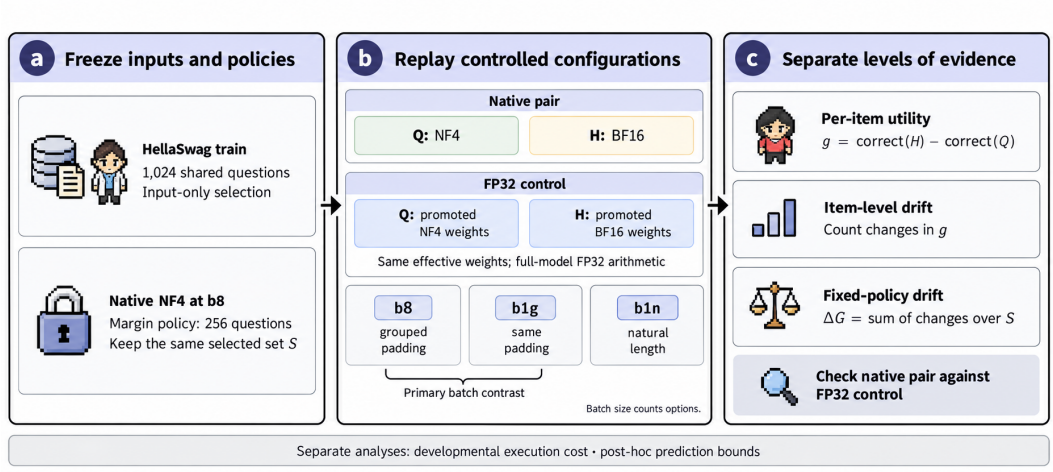

Figure 1: Conceptual audit workflow. Both models share the input panel. For each model and score, native NF4 reference outputs determine a label-free 256-question selection, which is held fixed across layouts and precision controls. The primary contrast changes option batch size while preserving padding. Each FP32 arm uses the effective weights of its corresponding native arm. Developmental cost measurements and the post-hoc bound form separate analyses.

two distillation objectives while tracking fit quality, evaluation fidelity, and rescue–harm balance. The accompanying protocol and saved-output analyses preserve these distinctions, including the chronology of confirmation and subsequent diagnostics.

2 Paired utility and the evaluation contract

Figure 1 organizes the audit into input and policy selection, controlled execution, and measurements at different levels. The precision control preserves each arm's own effective weights while changing its arithmetic to FP32.

Let $Q_{i,\ell}$ and $H_{i,\ell}$ be the predicted option indices from low-precision and higher-precision paths on example i under execution layout ℓ. With gold label y_i, define

$$g_{i,\ell} = \mathbf{1}\{H_{i,\ell} = y_i\} - \mathbf{1}\{Q_{i,\ell} = y_i\} \in \{-1, 0, 1\}. \tag{1}$$

Values $+1$ and -1 correspond to rescue and harm. For a fixed selection S, the policy's gain over its corresponding Q-only baseline is

$$G_\ell(S) = \sum_{i \in S} g_{i,\ell}, \qquad \Delta G(S) = G_b(S) - G_a(S). \tag{2}$$

Gains are measured in integer net-correct units relative to the Q-only baseline at the same layout. The policy's total correct count also includes that baseline's correct answers.

Writing $C_\ell^Q = \sum_i \mathbf{1}\{Q_{i,\ell} = y_i\}$ and $C_\ell(S)$ for the policy's total correct count gives the exact decomposition

$$C_\ell(S) = C_\ell^Q + G_\ell(S), \qquad \Delta C(S) = \Delta C^Q + \Delta G(S). \tag{3}$$

Here, $C_\ell(S)$ describes an offline composite of the fixed-layout prediction tables. Equation 3 separates changes in this total from changes in the advantage over the corresponding Q-only baseline. Actual fallback execution is measured separately in Section 4.4.

We measure the whole-panel count $U = \sum_i \mathbf{1}\{g_{i,a} \neq g_{i,b}\}$, its selected-set counterpart U_S, and the signed aggregate $\Delta G(S)$. Two selected examples with changes $+1$ and -1 contribute two to U_S but zero to ΔG. An individual change can also have magnitude two.

What is held fixed. The primary selection contains 256 examples with the smallest native NF4 margins at the reference layout, using the prespecified score-softmax and tie rules. This set is transferred unchanged to all layouts and to the FP32 weight control. We also report an entropy policy. Both policies use score-softmax to rank length-normalized continuation scores; the resulting values are uncalibrated ranking quantities. Appendix B separately examines layout-specific reselection.

The audit selects the highest-ranked quarter of the complete panel at a fixed budget. Under uniform random selection of k questions, linearity of expectation gives $\mathbb{E}[G_\ell(S)] = (k/N)\sum_i g_{i,\ell}$. This analytic expectation supplies the random-selection comparator alongside margin, entropy, and both single-precision endpoints. The developmental cost replay instead uses a threshold fitted on development inputs.

Execution cost. Executing fallback can create high-precision sub-batches with different shapes from those used in the offline prediction table. These changes can affect both predictions and cost. We therefore measure latency and memory directly in the developmental replay described in Section 4.4.

3 FROZEN CONFIRMATION DESIGN

The design crosses two models, four precision paths, three layouts, and two exact repeats on one shared panel. The primary comparison changes batch size while preserving each option's padded input. Questions are the analysis units. Appendix A details panel construction, scoring, and selection.

Models and data. We use Qwen2.5-1.5B-Instruct and SmolLM2-1.7B-Instruct (Qwen et al., 2024; Ben Allal et al., 2025). Both models evaluate 1,024 examples from the HellaSwag training corpus (Zellers et al., 2019). Selection ranks inputs by a fixed hash and excludes previously used source IDs and exact processed queries before tokenization. The procedure uses input text and identifiers, independently of gold labels and model outcomes. Maximum input lengths are 142 and 154 tokens for Qwen and Smol, respectively. All selected inputs are retained at their original lengths. The panel is source-disjoint from the earlier developmental panel within the same training corpus.

Scoring and layout. Each option is scored by its full continuation log likelihood with causal alignment. The primary score uses explicit FP32 log-softmax and summation, normalized by the processed continuation's Python character count. Three additional variants cross unnormalized scoring with inherited-output-dtype postprocessing. Our controlled evaluator follows the pinned LM Evaluation Harness task text, token-boundary, and normalization rules (EleutherAI, 2026).

The three layouts are grouped-padding option batch 8 ($b8$), identical grouped padding with option batch 1 ($b1g$), and natural-length option batch 1 ($b1n$). The primary contrast is $b1g - b8$, isolating batch size while retaining each option's padding. Batch size counts options, not questions. The forward call uses right padding with token ID zero, disables the key/value cache, and supplies neither an attention mask nor position IDs. The natural-length contrast is secondary because it changes padding as well.

Precision paths. The native pair uses the original BF16 model as H and bitsandbytes NF4 with BF16 compute as Q. We use the NF4 format introduced by Dettmers et al. (2023) for inference without adapter training. Each control arm promotes its corresponding native arm's effective weights to FP32 and executes full-model FP32 arithmetic. Weight values and shared parameters are verified against the source records. This promotion preserves existing weight information, including quantization loss, while changing arithmetic throughout the model. The output-only FP32 scoring operation is common to the primary score in all four paths.

Decision rule and repetitions. The predeclared continuation rule requires both models to satisfy two criteria. C1 requires at least 11 native examples with changed g and at most one in the FP32 control. C2 requires native $|\Delta G| \geq 11$ and control $|\Delta G| \leq 1$ for the fixed primary policy. Eleven net-correct units correspond to 1.0742 percentage points on this panel. This threshold governs continuation of the research program; it is not a statistical significance threshold. Primary results are descriptive counts on the selected panel. Each mode-layout combination is repeated twice to check exact execution consistency. The resulting 196,608 option-execution records represent 1,024 distinct questions, with the first repeat supplying the reported counts.

4 RESULTS

4.1 ITEM-LEVEL CHANGES LARGELY CANCEL WITHIN THE FIXED POLICY

Both native precision pairs change item-level upgrade utility under the primary layout contrast, whereas the FP32 controls have no changed utility labels (Figure 2; exact summary in Table 1). The two repetitions agree exactly within each fixed configuration. Thus, repeat consistency and cross-layout utility stability answer different questions.

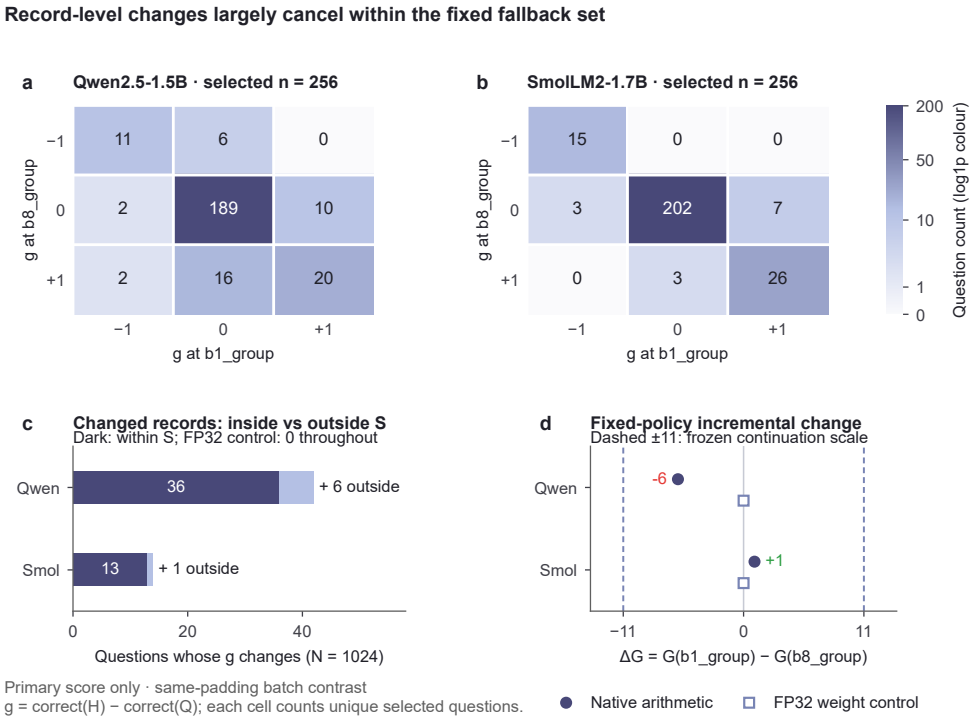

Figure 2: Item-level changes and aggregate drift answer different questions. **a,b**, Transitions from reference-layout g (rows) to same-padding batch-1 g (columns), restricted to each model's fixed 256-question margin selection. Values $-1, 0, +1$ denote harm, equal Q/H correctness (both correct or both incorrect), and rescue. Colour uses a shared $\log(1 + \text{count})$ mapping; printed counts are exact. **c**, Changed questions inside and outside the selected set, among all 1,024 questions. **d**, Signed fixed-policy gain change for native and FP32-control pairs; dashed ± 11 lines mark the frozen management scale, not confidence limits. Both models use instruction-tuned checkpoints. Panels use the primary score and first repeat; repeat two checks integrity.

Table 1: Frozen primary confirmation on 1,024 shared questions. U counts whole-panel changes in g; U_S restricts to the fixed margin selection of 256 questions. Parenthesized values are FP32 controls.

Model	U	U_S	$G_{b8} \to G_{b1g}$	$100\Delta G/N$ (pp)	C1 / C2
Qwen	42 (0)	36	$21 \to 15$	-0.586 (0)	pass / fail
Smol	14 (0)	13	$14 \to 15$	$+0.098$ (0)	pass / fail

Within Qwen's selected set, two examples change by -2, eighteen by -1, and sixteen by $+1$. These 36 changes sum to -6 net-correct units. Smol has six changes of -1 and seven of $+1$, summing to $+1$. Both models satisfy C1, while aggregate drift remains below C2's 11-unit continuation scale.

Signed cancellation. A post-hoc decomposition quantifies cancellation in the recorded changes. Qwen has 16 positive and 22 negative net-correct units of change; Smol has seven and six. Opposing

signs cancel 84.2% and 92.3% of the absolute change, respectively (Appendix B). Off-diagonal matrix entries count changed examples, whereas their signed magnitudes determine aggregate drift.

Equation 3 also matters in practice. Qwen's Q-only correct count changes from 688 to 687, so its policy changes from 709 to 702: baseline drift of -1 plus incremental drift of -6. Smol's Q-only count changes from 733 to 731, while its policy changes from 747 to 746. Its incremental gain therefore increases by one even though its total correct count decreases by one. This distinction is not captured by either the instability count or the sign of ΔG alone.

4.2 Margin selection and the rescue–harm balance

The reference-layout results in Figure 3 include both rescue and harm, using the same native selections for the FP32 control. Rescue and harm persist under FP32 arithmetic even though their utility labels remain stable across layouts. The comparison therefore separates differences between the two effective-weight predictors from layout-dependent changes in their paired utility.

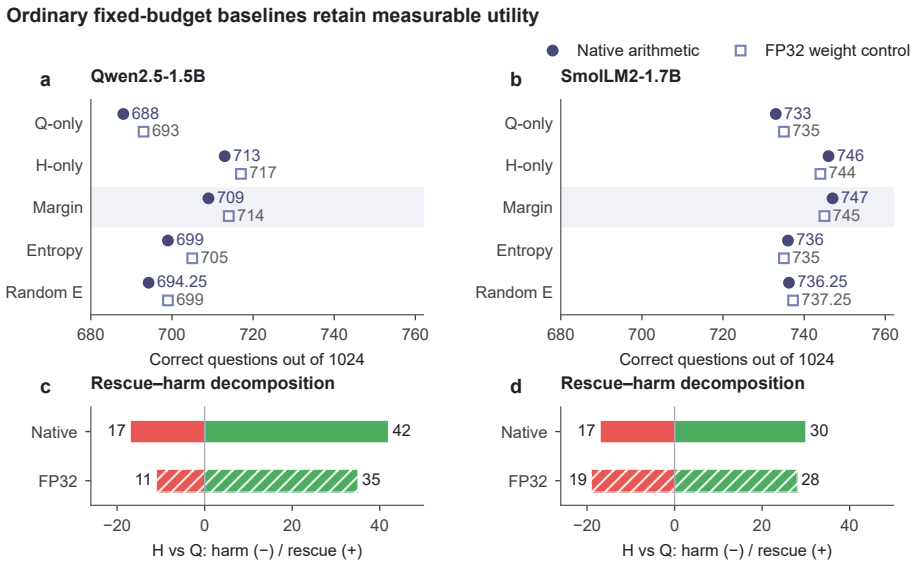

Figure 3: Fixed-budget baselines and their precision-pair context. **a,b**, Correct counts on the same 1,024 questions at the primary $b8$ layout. Margin and entropy select 256 questions; random denotes their analytic same-budget expectation. H-only evaluates all questions at higher precision. The native-selected sets are transferred unchanged to the FP32 control. **c,d**, Whole-panel rescue and harm counts for H relative to Q, not only within the selected set. Harm appears left of zero; counts themselves are nonnegative. Both models use instruction-tuned checkpoints. These finite-panel correctness measurements are not equal-cost comparisons or latency estimates.

Margin selection exceeds entropy in both models and precision pairs. Smol's native margin policy yields 747 correct answers, compared with 746 for H-only. Selective fallback can retain correct Q predictions that would be harmed by an upgrade, so its total can exceed the higher-precision endpoint. Appendix E reports the eight-gate learned-probe comparison on ARC.

4.3 Scoring choices alter measured layout effects

All four predeclared scoring variants appear in Appendix D, with common visualization scales in Figure 7. Qwen's inherited-dtype, unnormalized variant yields $\Delta G = +11$, compared with $+1$ for Smol. The primary analysis follows the prespecified FP32 character-normalized score. Across the reported fixed policies, continuation scoring produces no strictly positive-to-negative utility reversal.

A separate developmental study compared evaluation recipes on 320 previously observed HellaSwag questions. Smol's native BF16 correct count changed from 80 under custom answer-letter scoring to 237 under character-normalized continuation scoring. Its NF4 count changed from 85 to 224. Prompt presentation, scoring, and option execution changed together, so this comparison measures the combined recipe difference. The main experiment uses the continuation recipe fixed before collection of its new-panel outputs.

4.4 Execution cost in the developmental replay

Earlier replay measurements used a different task and recipe: 600 ARC-Challenge examples (Clark et al., 2018) with answer-letter scoring and a frozen entropy gate. At batch 8, direct BF16 took 2.1866 seconds with 78.0% accuracy, whereas native NF4 plus fallback took 4.8617 seconds with 75.6667% accuracy. Post-warmup PyTorch peak allocated memory was 3.0001 versus 4.1316 GiB. The fallback gate selected 150 examples, but its executed high-precision sub-batches changed nine predictions relative to the offline BF16 reference. Consequently, the offline policy estimate of 76.1667% did not equal the executed 75.6667%.

Times are medians of three complete sequential replays, measured from tokenized CPU inputs to CPU predictions. They include transfers, gating, fallback, and output fusion, and exclude tokenization, cold model loading, and result-file I/O.

Figure 9 reports all three batch sizes. At every matched batch size, fallback is slower, less accurate, and uses more peak allocated memory than direct BF16 in this implementation. The replay exposes a difference between reducing high-precision calls and reducing measured execution cost.

4.5 Improved fitting and evaluation utility diverge under repair

The repair diagnostics correct the final representation before the output head; utility compares corrected and uncorrected Q predictions. A rank-16 reduced-rank ridge fit lowers hidden-state MSE by 19.03% on a 256-question holdout. A separately fitted full-training correction lowers it by 19.43% on the observed transfer panel, while correct answers decrease from 736 to 729. Centered-score MSE improves by 3.99% and 5.11%, respectively. The learned residual captures cross-question representation structure, but its average reconstruction improvement and answer utility move differently (Appendix G).

We then compare hidden MSE, centered-score MSE, option KL, and token KL using the same 65,536-parameter correction on SmolLM2. All four fit the same 768 questions from the same initialization; 256 questions are held out from fitting, and the transfer panel contains 1,024 previously observed questions. Both model paths execute their respective effective weights in FP32. Updates use teacher targets without task labels. Option KL and token KL provide standard distillation controls (Hinton et al., 2015). The centered-score objective is designated before this comparison, and the four final states are evaluated without checkpoint selection on either evaluation panel.

Adaptive backtracking accepts 32 updates for every objective. Centered-score fitting reduces its training MSE by 46.17%, while validation and transfer MSE increase by 15.24% and 8.16% (Figure 4). On the transfer panel, it rescues 14 answers and harms 11, yielding 739 correct answers; option KL yields 741, and token KL yields 736. Agreement with the teacher also differs from average score fidelity: centered-score fitting increases transfer disagreements from 57 to 63. The optimizer reaches the accepted-update cap in every arm. Appendix G reports all objectives, the preceding fixed-radius diagnostic, and the observed optimization trajectories.

5 Interpretation and limitations

Measure utility at the intervention's endpoint. For selective precision, measured benefit depends on the scoring rule, selected set, and execution configuration. Individual outcome changes, their signed aggregate within a fixed policy, and the cost of executing that policy therefore belong in the same evaluation. The main experiment measures the first two under controlled continuation scoring; the ARC replay measures executed outcomes and cost under its answer-letter recipe.

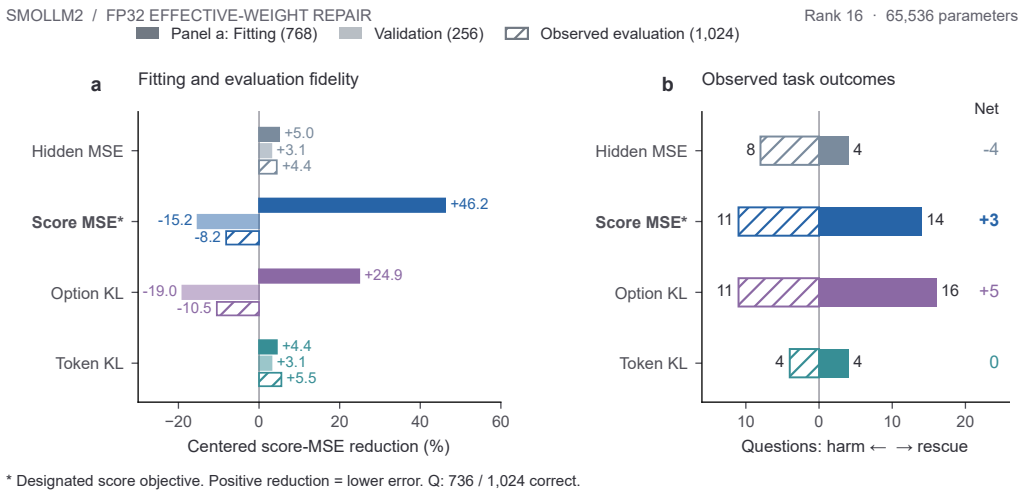

Figure 4: Fitting, evaluation fidelity, and task utility under adaptive repair. **a**, Centered-score MSE reduction relative to Q across the 768-question fit, 256-question validation, and observed 1,024-question transfer panels. Positive values denote lower error; the top legend applies to this panel. **b**, Transfer rescues (solid, right) and harms (hatched, left) relative to Q's 736 correct answers, with net differences at right. The asterisk marks the designated score objective. Each row uses one final correction with the same architecture and fitting split; counts describe the previously observed panels.

Item-level upgrade utility is sensitive to layout, yet opposing changes largely cancel inside the fixed policy. Reporting both quantities reveals which examples change and how those changes affect the selected policy. This connects prediction-flip measurements to policy-level evaluation, building on studies of quantization and numerical reproducibility.

Repair adds a further distinction between fitting a teacher-target objective and improving discrete answers. Hidden-state reconstruction reduces average residual error on evaluation inputs, whereas centered-score fitting concentrates its gains on the fitting split. Neither quantity determines the rescue–harm balance by itself. Layerwise reconstruction methods such as LQER and QERA target quantization errors throughout the network; the final-state diagnostic here isolates a common correction location and several teacher-target objectives. This comparison identifies which measurement changed under a fixed intervention, rather than assigning the outcome to reconstruction loss alone.

Scope. The measured scope is multiple-choice evaluation of two small instruction-tuned models using one NF4 implementation and three layouts; repair uses SmolLM2, one correction rank, and a finite optimization budget. The public training-corpus panels do not resolve pretraining contamination or population-level equivalence. The scoring evaluator implements the pinned task recipe, and the cost results concern the ARC replay. Native low-bit repair deployment, larger models, and long-form generation require their own evaluation.

Analysis chronology. The two-model panel and continuation criteria were fixed before collecting its model outputs; the joint continuation rule stopped after C2 was unmet. The synthesis, cancellation decomposition, and prediction-only bound followed those observations. The later repair series used previously observed panels, with its data split, objectives, and adaptive-update rule fixed before the four-arm run. Those results are descriptive diagnostics; exact repeats check execution consistency. All adaptive repair arms stopped at their accepted-update limit, leaving convergence and the relative roles of objective, parameterization, and data quantity unresolved.

6 CONCLUSION

The controlled audit separates prediction changes from task utility at three points: aggregation within a fixed fallback set, execution of the selected policy, and transfer of teacher-target fitting to discrete answers. Layout-sensitive changes largely cancel inside the evaluated policies, offline fallback estimates differ from executed outcomes, and improved reconstruction or fitting does not imply higher task accuracy in the repair diagnostics. Measuring rescue, harm, and realized cost alongside stability and fidelity makes these differences explicit.

REPRODUCIBILITY STATEMENT

The accompanying materials document pinned revisions, input-only panel selection, saved outputs, scoring variants, and analysis procedures. Independent recomputation checked 345,617 fields across the two-model analyses and their joint decision. Repair materials include final scores, fitting splits, recorded optimization traces, and CPU recomputation of the reported metrics. Appendices A, F, and G specify the protocols and computational scope. Instructions distinguish saved-output analysis from fresh model inference and gradient computation.

AI USE STATEMENT

The authors used OpenAI Codex to assist with hypothesis refinement, experimental design, implementation and testing, mathematical formulation and checks, result interpretation, literature search, and manuscript drafting and editing. OpenAI's image-generation tool generated the two conceptual illustrations (Figures 1 and 5). Quantitative figures were plotted from recorded experimental data. The authors reviewed the AI-assisted content and take responsibility for the final manuscript. The supplementary material includes the illustration prompts, manuscript sources, and figure data.

ETHICS STATEMENT

This study uses publicly available benchmark datasets and pretrained language models. The supplementary material documents their sources and licenses. The analysis reports both beneficial and harmful prediction changes under precision upgrades and repair. These measurements inform the evaluation of selective precision in deployed systems, where task-specific accuracy, computational cost, and failure cases should be assessed together.

REFERENCES

Larry Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966. https://msp.org/pjm/1966/16-1/pjm-v16-n1-p01-p.pdf.

Loubna Ben Allal et al. SmolLM2: When Smol Goes Big – Data-Centric Training of a Small Language Model. arXiv:2502.02737, 2025. https://arxiv.org/abs/2502.02737.

Kexin Chu, Yang Zhou, and Wei Zhang. MarginGate: Sparse Margin-Triggered Verification for Batch-Invariant LLM Inference. arXiv:2605.30218v1, 2026. https://arxiv.org/abs/2605.30218v1.

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

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient Finetuning of Quantized LLMs. arXiv:2305.14314, 2023. https://arxiv.org/abs/2305.14314.

Abhinav Dutta, Sanjeev Krishnan, Nipun Kwatra, and Ramachandran Ramjee. Accuracy is Not All You Need. *Advances in Neural Information Processing Systems*, 37, 2024. https://doi.org/10.52202/079017-3950.

EleutherAI. Language Model Evaluation Harness. Software repository, accessed September 2026. Pinned implementation revision in Appendix F. https://github.com/EleutherAI/lm-evaluation-harness/tree/ad8737ae7fad24cf64e50fc7fc31397bff586b9e.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. arXiv:1503.02531, 2015. https://arxiv.org/abs/1503.02531.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685, 2021. https://arxiv.org/abs/2106.09685.

Qwen et al. Qwen2.5 Technical Report. arXiv:2412.15115, 2024. https://arxiv.org/abs/2412.15115.

Zekun Wu, Swati Dhiman, and Adriano Koshiyama. Which Decisions Low-Bit Quantization Breaks, and How to Predict Them. arXiv:2608.06564v3, 2026. https://arxiv.org/abs/2608.06564v3.

Jiayi Yuan, Hao Li, Xinheng Ding, Wenya Xie, Yu-Jhe Li, Wentian Zhao, Kun Wan, Jing Shi, Xia Hu, and Zirui Liu. Understanding and Mitigating Numerical Sources of Nondeterminism in LLM Inference. arXiv:2506.09501v2, 2025. https://arxiv.org/abs/2506.09501v2.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a Machine Really Finish Your Sentence? ACL, 2019. https://arxiv.org/abs/1905.07830.

Cheng Zhang, Jianyi Cheng, George Anthony Constantinides, and Yiren Zhao. LQER: Low-Rank Quantization Error Reconstruction for LLMs. *Proceedings of the 41st International Conference on Machine Learning*, PMLR 235:58763–58779, 2024. https://proceedings.mlr.press/v235/zhang24j.html.

Cheng Zhang, Jeffrey T. H. Wong, Can Xiao, George A. Constantinides, and Yiren Zhao. QERA: an Analytical Framework for Quantization Error Reconstruction. ICLR, 2025. https://arxiv.org/abs/2410.06040.

A Detailed evaluation protocol

A.1 Input-only construction of the confirmation panel

The source contains 39,905 HellaSwag training rows. Before ranking, the selector excludes any row matching a previously observed question's exact source ID, processed query, or processed query–ordered-choices identity. The overlapping filters match 539, 320, and 320 rows individually and exclude 539 rows in total. The earlier 320-question panel contains 318 distinct source IDs. Source-level exclusion also removes associated questions with different question IDs.

The remaining 39,366 rows are sorted by the SHA256 of a frozen seed concatenated with the stable row/ind identifier, followed by that identifier. The selector greedily retains a candidate only when both its source ID and its exact processed-query digest are new within the selected panel. It examines 1,033 ranked candidates, skipping nine repeated sources and zero additional repeated queries, to obtain 1,024 source- and query-unique questions. The seed and canonical digest rules are included in the retained artifact configuration. No gold-label field is accessed by this selector.

Tokenization occurs after this text-level panel is fixed. An over-length sequence, an empty continuation, or a failed token-prefix check stops execution without truncation or replacement. The cap is 1,024 input tokens. Qwen's 4,096 option inputs range from 18 to 142 tokens and contain 310,320 tokens in total; Smol's range from 17 to 154 and contain 312,031. These input-volume totals include repeated contexts across options; the question remains the scientific unit. The observed maxima are well below the cap.

A.2 Continuation score and deterministic selection

For processed option text v_{ij}, let L_{ij} be its continuation-token count and I_{ij} the input length after omitting only the final target token. If $Z_{ij,t}$ denotes the full-vocabulary logit vector, the primary score is

$$s_{ij} = \frac{1}{\mathrm{len}_{\mathrm{char}}(v_{ij})} \sum_{t=I_{ij}-L_{ij}}^{I_{ij}-1} \left[\log \mathrm{softmax}_{\mathrm{FP32}}(Z_{ij,t})\right]_{\mathrm{target}_{ij,t}} . \tag{4}$$

The denominator is Python's character count of the processed choice, not its token count; it excludes the additional leading space used in the continuation request. The causal slice includes the first continuation prediction at the last context position and excludes padding and context targets.

1. Apply the pinned task preprocessing to obtain one query and four original-order choices. Tokenize the query plus a leading-space continuation together, and verify that the tokenized query is an exact prefix.
2. Form context-plus-continuation inputs with the final target token removed. Build the declared right-padded layout without changing the order of options.
3. Execute the causal model in evaluation and inference modes. Apply full-vocabulary FP32 log-softmax to the complete forward logit tensor before slicing continuation positions.
4. Gather the target-token values, sum on the original device in that postprocessing dtype, and divide by the processed choice's character count. Compute inherited-dtype postprocessing from the same forward logits.
5. Select the option of maximum score, breaking exact ties by original option order. All 24 primary-score model–mode–layout cells have zero exact top-two ties.

The four-option score-softmax used for routing is computed separately in NumPy float64:

$$p_{ij} = \frac{\exp(s_{ij} - \max_k s_{ik})}{\sum_k \exp(s_{ik} - \max_k s_{ik})}, \quad r_i^{\mathrm{margin}} = p_{i,(2)} - p_{i,(1)}, \quad r_i^{\mathrm{entropy}} = -\sum_j p_{ij} \log p_{ij}, \tag{5}$$

where $p_{i,(1)} \geq p_{i,(2)}$. Sort each gate score in descending order, then by SHA256 of the question ID and the ID in ascending order, and keep 256 questions. The reference is native NF4 at $b8$ for the corresponding score. This label-free procedure ranks the full panel; it does not fit a request-level threshold. The score-softmax supplies ranking scores without a probability-calibration guarantee.

A.3 FORWARD AND PRECISION CONTROLS

All paths load from the same recorded BF16 checkpoint values and use eager attention. Native NF4 uses BF16 compute with double quantization disabled. The two FP32 controls promote the corresponding effective values, including the quantized path's dequantized weights, without reconstructing information absent from those values. Non-quantized parameters, buffers, and shared embedding/output-head states are checked against the source receipts.

TF32 is disabled in all four paths. The native default paths retain the recorded reduced-precision BF16 reduction setting; the strict FP32 paths disable it. The forward call supplies `input_ids` and `use_cache=False` and omits `attention_mask` and `position_ids`. Stored mask and position arrays serve as audit metadata only. The arithmetic path and this calling convention jointly define the evaluated execution.

For $b8$, each consecutive group of eight options is padded to its group's maximum input length. For $b1g$, each option is evaluated alone at that same recorded padded length. For $b1n$, each option is evaluated alone at its natural input length. Thus $b1g - b8$ preserves each option's token tensor while changing batch shape; $b1n$ also changes padding. Two repetitions of each cell are retained for exact integrity checking, with only repeat zero supplying scientific counts.

B CANCELLATION, TOTAL CORRECTNESS, AND RESELECTION

This section gives post-hoc decompositions of the recorded confirmation outcomes under the original decision rule. Figure 5 illustrates cancellation with a constructed example; Figure 2 reports the measured transfer matrices.

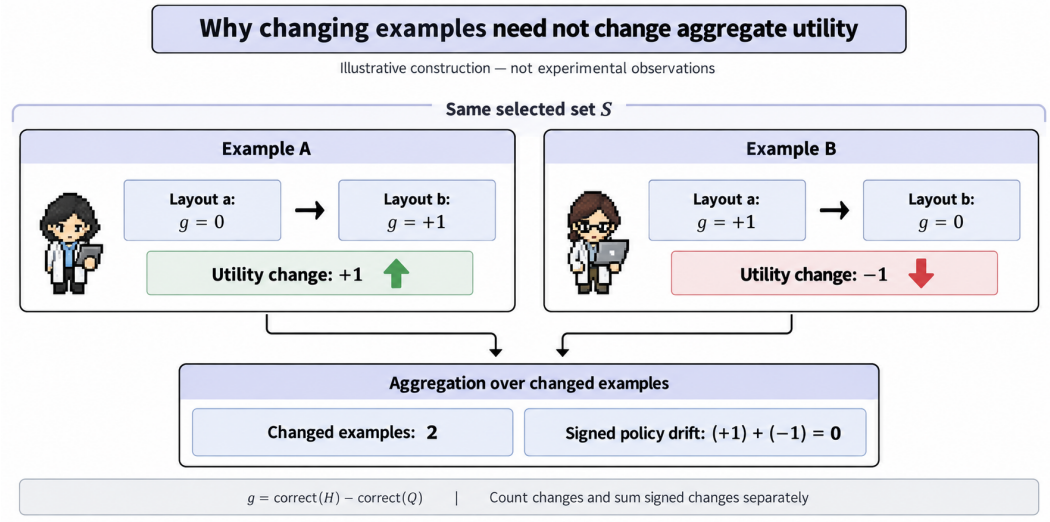

Figure 5: Constructed example of signed cancellation. Two selected records change utility in opposite directions. The changed-record count is two, whereas the signed aggregate change is zero. Record values give utility before and after the layout change; arrows give utility changes. Rescue describes a utility value of $+1$, distinct from a utility change of $+1$.

B.1 CHANGE COUNTS VERSUS CHANGE MAGNITUDE

With $\delta_i = g_{i,b} - g_{i,a}$, define positive and negative change mass by

$$P = \sum_{i \in S} \max(\delta_i, 0), \quad M_- = \sum_{i \in S} \max(-\delta_i, 0), \quad M = P + M_-, \quad \Delta G = P - M_-. \tag{6}$$

For $M > 0$, the descriptive cancellation fraction is

$$\kappa = 1 - \frac{|\Delta G|}{M} = \frac{2\min(P, M_-)}{P + M_-}. \tag{7}$$

The cancellation fraction is undefined for $M = 0$. M measures absolute net-correct units, whereas U_S counts changed questions: δ_i can be -2 or $+2$.

Table 2: Post-hoc decomposition of primary-score native changes under $b1g - b8$. Each gate uses its own frozen 256-question reference selection.

Model	Gate	P	M_-	M	ΔG	κ (%)
Qwen	Margin	16	22	38	-6	84.21
Smol	Margin	7	6	13	$+1$	92.31
Qwen	Entropy	8	10	18	-2	88.89
Smol	Entropy	4	4	8	0	100.00

B.2 Baseline and total-policy correctness

The offline composite policy uses the saved H prediction on S and the saved Q prediction elsewhere. Table 3 decomposes its total correctness into the layout-specific baseline and fixed-set incremental gain; C2 evaluates the latter's change.

Table 3: Total-policy correctness versus incremental gain on the same 1,024 questions. Counts are $b8 \to b1g$; the final column verifies Equation 3.

Model	C^Q	$G(S)$	$C(S)$	$\Delta C^Q + \Delta G = \Delta C$
Qwen	$688 \to 687$	$21 \to 15$	$709 \to 702$	$-1 - 6 = -7$
Smol	$733 \to 731$	$14 \to 15$	$747 \to 746$	$-2 + 1 = -1$

B.3 Fixed-set drift and reselection

If each layout instead selects a new set S_ℓ, its gain change contains two terms:

$$G_b(S_b) - G_a(S_a) = \underbrace{G_b(S_a) - G_a(S_a)}_{\text{fixed-set drift}} + \underbrace{G_b(S_b) - G_b(S_a)}_{\text{reselection term}}. \tag{8}$$

For primary native margin, Qwen's two sets overlap on 224 of their 256 selected questions (Jaccard 0.7778). Recomputing the batch-1 selection raises its batch-1 gain from 15 to 17, so the reselected contrast is $-6 + 2 = -4$. Smol's overlap is 249 (Jaccard 0.9468); reselection changes the batch-1 gain from 15 to 14, giving $+1 - 1 = 0$. These reselected offline policies have different selected sets; C2 retains the common reference set.

Selecting a new policy for the FP32 control also changes the comparison. Qwen's native and FP32 reference margin sets overlap on 222 questions. At the FP32 reference, the primary common native set yields gain 21, whereas the separately selected FP32 set yields 22. Holding the native set fixed isolates the execution contrast from this selection difference.

C POST-HOC PREDICTION-ONLY UTILITY BOUNDS

For already observed predictions from two layouts and a fixed selection, define

$$d_i(y) = \mathbf{1}\{H_{i,b} = y\} - \mathbf{1}\{Q_{i,b} = y\} - \mathbf{1}\{H_{i,a} = y\} + \mathbf{1}\{Q_{i,a} = y\}. \tag{9}$$

Let $D_- = \sum_{i \in S} \min_{y \in \mathcal{Y}_i} d_i(y)$ and $D_+ = \sum_{i \in S} \max_{y \in \mathcal{Y}_i} d_i(y)$. Then $D_- \leq \Delta G \leq D_+$ for every assignment of legal labels to these records. Both endpoints are attainable over the Cartesian product of legal labels by selecting an argmin or argmax separately for each record. This finite-record construction requires no statistical independence assumption.

For each record $\sum_{y \in \mathcal{Y}_i} d_i(y) = 0$, so every interval contains zero. The bound therefore describes possible drift magnitudes, rather than certifying positive drift or layout superiority without labels. Its inputs are both arms' predictions at the two observed layouts.

As comparators, let A count selected records where either arm changes its prediction, and let B sum the separate selected Q- and H-arm change counts. The triangle inequality gives

$$\max(|D_-|, |D_+|) \leq B \leq 2A. \tag{10}$$

If both arms move from the same answer to the same new answer, their differences cancel and d_i is identically zero despite changed predictions. The implementation counts this case and validates the exact extrema by exhaustive small-state tests.

The primary bounds in Figure 6 are only slightly tighter than the stronger armwise comparator for Qwen and identical to it for Smol. Two changed Qwen records have zero coefficient vectors; Smol has none. Neither native interval excludes an absolute change of 11, so the diagnostic cannot certify the observed small drift without labels. All 64 combinations of four scores, two precision pairs, two fixed gates, two contrasts, and two models are retained in the post-hoc output.

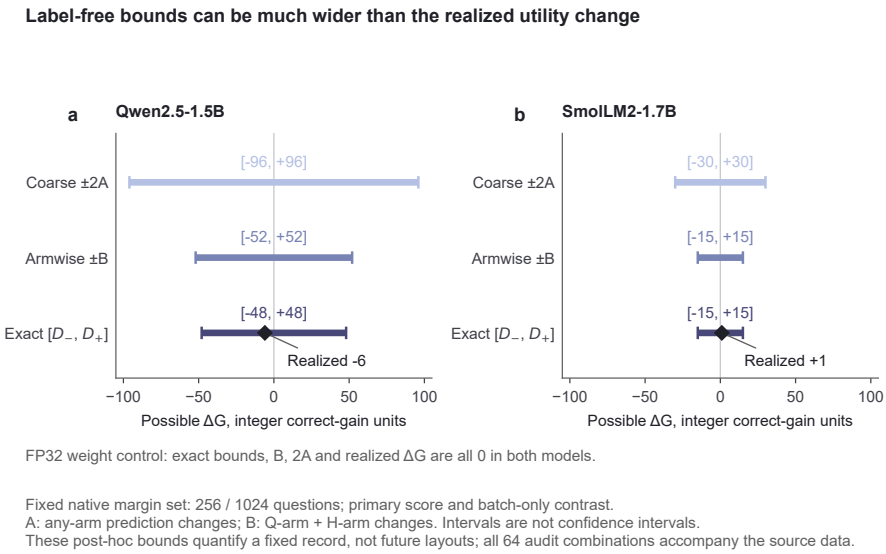

Figure 6: Prediction-only limits for the primary contrast and fixed selection. Exact endpoints, armwise limits $\pm B$, and coarse limits $\pm 2A$ concern the same observed records from the instruction-tuned models. Points show previously observed gold-based drift, checked against the bounds. These are finite-record limits, not confidence intervals. Both FP32 controls collapse to zero; neither native bound certifies drift smaller than 11.

D ALL PREDECLARED SCORE VARIANTS

Table 4 reports all prespecified scoring alternatives. Native U_{union} counts the union across all three layouts; all corresponding FP32 U counts and ΔG values are zero. In the scoring labels, native denotes postprocessing in the inherited output dtype. Internal arithmetic follows the precision settings in Appendix A.

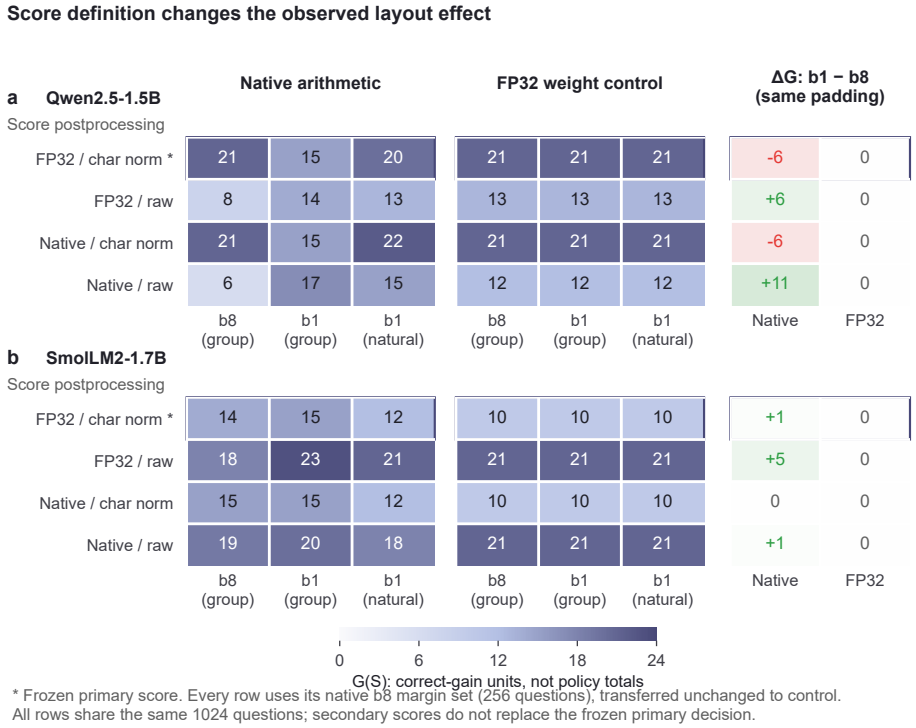

Figure 7: Prespecified score and layout comparison for the instruction-tuned checkpoints. Each score uses its own fixed native-reference margin selection, transferred unchanged across layouts and precision pairs. Heatmaps show net-correct gain on a common colour scale; the adjacent signed comparison isolates $b1g - b8$. The primary score is marked explicitly. Natural-length batch 1 changes both batch size and padding.

Table 4: Descriptive score variants. Gain triplets use the corresponding fixed native margin selection and the order $b8/b1g/b1n$. FP32 G uses that same set, not an independently selected FP32 set.

Model	Score	$U_{\text{batch}}/U_{\text{union}}$	Native G triplet	ΔG	FP32 G
Qwen	FP32 normalized (primary)	42/51	21/15/20	-6	21
Qwen	FP32 unnormalized	33/34	8/14/13	$+6$	13
Qwen	Native normalized	42/52	21/15/22	-6	21
Qwen	Native unnormalized	34/37	6/17/15	$+11$	12
Smol	FP32 normalized (primary)	14/16	14/15/12	$+1$	10
Smol	FP32 unnormalized	7/11	18/23/21	$+5$	21
Smol	Native normalized	11/14	15/15/12	0	10
Smol	Native unnormalized	9/14	19/20/18	$+1$	21

E DEVELOPMENTAL GATE AND EXECUTION STUDIES

The ARC probe study trained on 200 development questions and evaluated on a previously observed 600-question pilot. A learned local-response probe obtained 460 correct answers versus 459 for the uncertainty-feature Ridge baseline, with nine paired improvements and eight regressions. Its conditional bootstrap interval included zero. This developmental comparison preceded the main confirmation.

What the eight gates measure. Four Ridge gates predict $\mathrm{correct}(\mathrm{BF16}) - \mathrm{correct}(\mathrm{NF4})$ using development-only standardization, an unpenalized intercept, and fixed penalty $\alpha = 10$. The 200 development questions contain 18 rescue and 11 harm events. The uncertainty baseline has six features: normalized four-choice entropy, negative margin, maximum probability, valid-letter probability mass, normalized full-vocabulary entropy, and normalized log input length. Probe, sham, and null variants append five response features: normalized Jensen–Shannon divergence, entropy change, probability change at the original winner, signed-margin change at that winner, and an argmax-flip indicator.

The probe recomputes the final decoder's last-token MLP down projection with checkpoint BF16 weights and captured NF4 activations. The correction is anchored to actual NF4 logits through the difference between corrected and reference final-normalization/head evaluations. Sham applies one shared, fixed Rademacher sign vector to the hidden-coordinate correction. Null uses once-dequantized NF4 weights in the same BF16 recomputation path; either control can produce a nonzero response. The remaining four gates rank entropy or negative margin before or after the true probe. All eight select 150 pilot questions with deterministic score and ID tie-breaking. The cost replay below instead uses a fitted threshold.

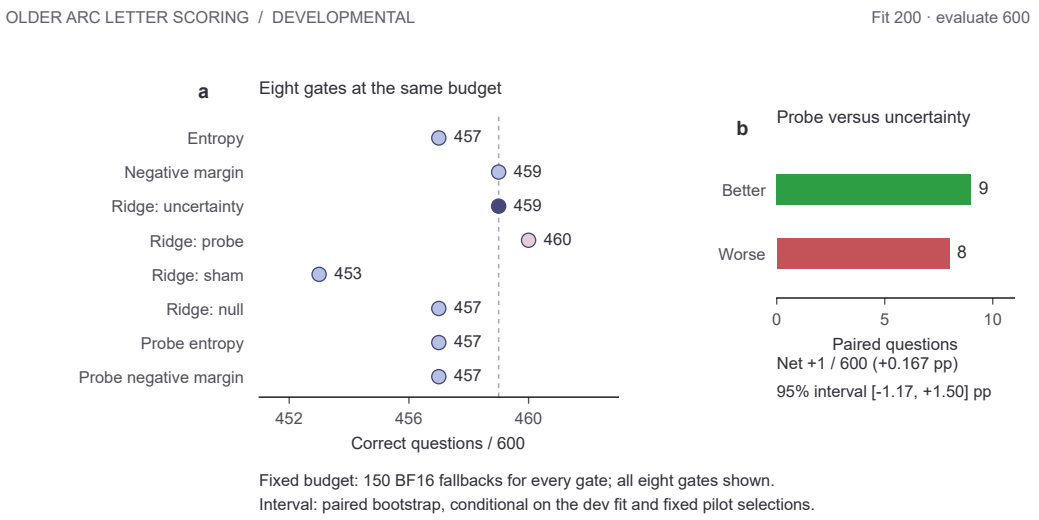

Figure 8: Developmental gates on the ARC answer-letter recipe. **a**, Correct counts from eight fixed gates, each selecting 150 of 600 pilot questions. **b**, The probe-versus-uncertainty Ridge comparison changes correctness on 17 questions, with nine improvements, eight regressions, and 583 unchanged. Its 95% paired-question bootstrap interval is $[-1.1667, +1.5000]$ percentage points from 2,000 resamples at the frozen seed. Resampling conditions on the fitted 200-question development panel and original selections, holding the fit and ranking fixed.

The 320-question scoring-recipe comparison also reused previously observed HellaSwag examples. Some fixed policies changed gain sign across layouts under custom answer-letter scoring. Continuation scoring produced no strictly positive-to-negative reversal. These developmental comparisons informed the protocol used for the main experiment.

E.1 ACTUAL REPLAY AT ALL MEASURED BATCH SIZES

Figure 9 reports nine strategy–batch settings, each measured with three complete sequential replays on the same 600 ARC questions. Runtime is the median, and memory is post-warmup PyTorch peak allocation. Timing includes transfers, gating, fallback sub-batches, and result fusion, and excludes tokenization, cold loading, and output-file I/O. These measurements account for the execution shapes created by fallback.

This replay applies an entropy threshold fitted to the original 200 development questions without labels. The fixed threshold triggers 140, 150, and 149 fallbacks at batch sizes 1, 8, and 32, respectively. Selected high-precision inputs retain their original padded positions while forming a smaller sub-batch.

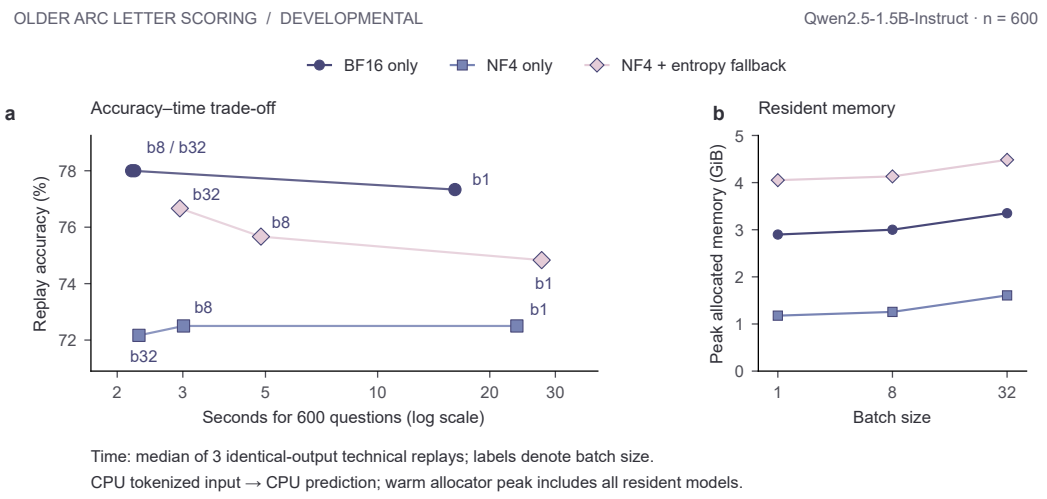

Figure 9: Developmental execution cost on 600 ARC questions. **a**, Actual correctness and median replay time for three strategies at batch sizes 1, 8, and 32. Points show measured values without jitter; nearby BF16 batch-8 and batch-32 points share a label. **b**, Recorded peak allocated memory. Each strategy is re-scored from saved predictions across three technical replays of the same questions. Entropy fallback is slower, less accurate, and uses more allocated memory than direct BF16 at every matched batch in this implementation.

Table 5: Exact replay measurements underlying Figure 9. H calls count questions evaluated at higher precision, including all questions for BF16-only. Correctness is out of 600; time is the median of three technical replays. Peak allocation includes all resident policy models.

Batch	Strategy	Correct	H calls	Time (s)	Peak (GiB)
1	BF16-only	464	600	16.1266	2.8979
1	NF4-only	435	0	23.6467	1.1769
1	NF4 + fallback	449	140	27.5665	4.0516
8	BF16-only	468	600	2.1866	3.0001
8	NF4-only	435	0	3.0126	1.2562
8	NF4 + fallback	454	150	4.8617	4.1316
32	BF16-only	468	600	2.2281	3.3504
32	NF4-only	433	0	2.2874	1.6065
32	NF4 + fallback	460	149	2.9451	4.4830

F IMPLEMENTATION AND PROVENANCE

The main runs used PyTorch 2.7.1, Transformers 4.51.3, bitsandbytes 0.45.5, and NumPy 1.26.4. Runtime metadata identifies both devices as NVIDIA GeForce RTX 4090 and reports 48,510 MB of device memory. The two collection runs took 2,330.60 and 1,884.23 seconds, with peak PyTorch allocated memory of 7.744 and 7.500 GiB. Per-replay timing is reported separately in Appendix E.

The exact revisions are listed below.

Component	Revision
Qwen checkpoint	`989aa7980e4cf806f80c7fef2b1adb7bc71aa306`
Smol checkpoint	`31b70e2e869a7173562077fd711b654946d38674`
Dataset	`218ec52e09a7e7462a5400043bb9a69a41d06b76`
Task reference	`ad8737ae7fad24cf64e50fc7fc31397bff586b9e`

Project provenance retains prelaunch hashes for 54 protocol, source, test, and reference files, together with the recorded selection-to-launch chronology.

The Linux and Windows Smol analyses and two-model decision summary are byte-identical. Qwen differs in one descriptive postprocessing scalar by 5.55×10^{-17}. All other structured fields, including predictions, selected sets, counts, and decisions, agree exactly. Both original analyses and the comparison record are retained in the project provenance.

Figure provenance. Quantitative figure scripts validate input hashes, match predictions by question ID, reconstruct fixed selections, and export machine-readable source data with vector and raster figures. Matrix totals equal the 256 selected questions, and signed changes reconcile with stored gain and policy-correct counts. Developmental replay plots independently re-score 16,200 saved prediction rows across nine settings and three technical replays. Source hashes and figure-level checks accompany the exports. The generated conceptual illustrations depict the evaluation design and a constructed cancellation example.

Rebuilding the figures and manuscript. Separate Python entry points rebuild the main confirmation and developmental data figures from saved results without GPU inference. Figure exports include editable-text SVG, embedded-font PDF, 300-dpi PNG previews, and compressed 600-dpi TIFF. LaTeX uses PDFs for data panels and generated PNGs for conceptual panels. Manuscript sources, illustration prompts, plotted data, and build instructions accompany the figures.

G TEACHER-TARGET REPAIR: PROTOCOL AND COMPLETE DIAGNOSTIC RESULTS

G.1 CORRECTION AND FOUR OBJECTIVES

The repair series uses the recorded SmolLM2 checkpoint and HellaSwag preprocessing. Let $x_{ijt} \in \mathbb{R}^d$ be a row-vector representation after final normalization for continuation token t of option j on question i, with $d = 2048$. The correction is

$$\widetilde{x}_{ijt} = x_{ijt} + (x_{ijt}A)B, \qquad A \in \mathbb{R}^{2048\times 16}, \quad B \in \mathbb{R}^{16\times 2048}. \tag{11}$$

This is a low-rank residual parameterization (Hu et al., 2021) applied at one fixed representation location. The two factors contain 65,536 trainable scalars. Both model paths retain their own effective weights and use full-model FP32 arithmetic; the frozen shared output head maps the corrected representation to the full vocabulary. The continuation score follows Equation 4, with original option batches of eight. The teacher supplies representations and scores, and gold labels are used for evaluation after fitting.

Write $s_i, s_i^H \in \mathbb{R}^4$ for the corrected and teacher continuation-score vectors, and $C = I - \frac{1}{4}\mathbf{1}\mathbf{1}^\top$. For the $n = 768$ fitting questions, the four targets are

$$L_{\text{hidden}} = \frac{1}{4n}\sum_{i,j}\frac{1}{dL_{ij}}\sum_t \|\widetilde{x}_{ijt} - x^H_{ijt}\|_2^2, \tag{12}$$

$$L_{\text{score}} = \frac{1}{4n}\sum_i \|C(s_i - s_i^H)\|_2^2, \tag{13}$$

$$L_{\text{option}} = \frac{T^2}{n}\sum_i \mathrm{KL}\big(\mathrm{softmax}(s_i^H/T) \,\|\, \mathrm{softmax}(s_i/T)\big), \quad T = 0.1, \tag{14}$$

$$L_{\text{token}} = \frac{1}{4n}\sum_{i,j}\frac{1}{L_{ij}}\sum_t \mathrm{KL}\big(p^H_{ijt} \,\|\, p_{ijt}\big). \tag{15}$$

Here L_{ij} is the continuation-token count and p_{ijt} is the full-vocabulary token distribution at temperature one. Question and option weights are equal; longer continuations are averaged within each option. Option and token KL are teacher-to-student distillation objectives (Hinton et al., 2015). Centering removes a common per-question score shift. Log-softmax and gathered continuation-score sums use FP32; objective aggregation, including token-KL reduction, uses FP64. The token-KL derivative retains the mass of rounded teacher probabilities.

The fixed 768/256 partition uses question-ID hashing and reuses the original SmolLM2 confirmation panel after its outcomes were observed. All four objectives share initialization ($A = A_0$, $B = 0$). A distinct 1,024-question transfer panel had also been observed before this comparison. Its uncorrected Q and teacher H answer 736 and 749 questions correctly; the fitting and validation counts are 555/561 and 180/183. Neither evaluation panel determines updates or checkpoint selection. The numerical paths are the effective-weight FP32 controls, rather than the native paths used for the main confirmation counts.

G.2 ANALYTICAL RECONSTRUCTION REFERENCE

The preceding reconstruction diagnostic fits a rank-16 reduced-rank ridge map from final Q representations to cumulative teacher residuals, with a constant mean-residual correction as comparator. Two fits were specified: one on 768 questions for the 256-question holdout, and one on all 1,024 fitting questions for the transfer panel. Both fits were run in the fixed order. Table 6 retains this parameter distinction.

Relative to the mean-residual comparator, the ridge fit lowers hidden MSE by 7.96% and 8.60% on these two panels. Mean correction yields 180 and 733 correct answers, compared with 178 and 729 for ridge. These observations concern cumulative final-state residuals; LQER and QERA instead construct quantization corrections at individual layers (Zhang et al., 2024; 2025).

Table 6: Reduced-rank ridge reconstruction. Reductions are relative to uncorrected Q error. The transfer row uses the separate 1,024-question fit. Teacher disagreements compare argmax answers, not gold correctness.

Evaluation / fitted questions	Hidden MSE ↓	Score MSE ↓	Disagreements	Correct
Validation / 768	19.03%	3.99%	$19 \to 20$	$180 \to 178$
Transfer / 1,024	19.43%	5.11%	$57 \to 61$	$736 \to 729$

G.3 Fixed-radius and adaptive optimization

The first four-objective comparison uses a fixed set of trial radii. It accepts 3/1/1/3 updates for hidden MSE, score MSE, option KL, and token KL before subsequent fixed-radius trials fail. Transfer net gains are $-2/0/-2/-1$. The adaptive comparison restarts all four objectives from the shared initialization and changes the search rule while preserving the fitting split and objective implementation.

The adaptive search uses Armijo backtracking (Armijo, 1966) with direction $-g/\|g\|_2$ for the selected fitting objective. Radius starts at 0.25, halves on rejection down to 2^{-22}, and the next iteration starts at twice the accepted radius, capped at 0.25. For the actual FP32 candidate θ', compute $P = -g^\top(\theta' - \theta)$ in FP64. Accept the first candidate satisfying $P > 0$, actual objective decrease at least $10^{-4}P$, and strict decrease greater than $10^{-8}\max(L_0, 10^{-12})$. Complete failure of a search stops at its current origin. The gradient stopping threshold is $\max(10^{-30}, 10^{-6}\|g_0\|_2)$.

The budget caps each arm at 32 accepted updates and 160 optimization panels, with an initial panel, a final gradient diagnostic, a transfer panel, and one fresh full-model check accounted for separately. All four objectives and their gradients are recorded by the shared diagnostic evaluator; only the designated objective drives that arm's updates. This instrumentation supplies a common measurement procedure, rather than each objective's minimal training cost.

G.4 Complete adaptive outcomes and stopping state

Table 7: Final adaptive corrections. Fit, validation, and transfer panels contain 768, 256, and 1,024 questions. MSE changes are percentage reductions in centered-score error relative to Q; negative values indicate increased error.

	Score MSE reduction (%)			Net correct answers		
Objective	Fit	Validation	Transfer	Fit	Validation	Transfer
Hidden MSE	4.99	3.05	4.40	−5	−1	−4
Score MSE	46.17	−15.24	−8.16	+3	−3	+3
Option KL	24.86	−19.02	−10.47	+6	+1	+5
Token KL	4.42	3.09	5.55	+1	−1	0

Teacher-disagreement counts on fit/validation/transfer are 48/21/59 for hidden MSE, 42/18/63 for score MSE, 30/15/67 for option KL, and 44/22/60 for token KL. The baseline counts are 44/19/57. Score fitting has lower evaluation MSE than option KL, but more validation disagreements and fewer net-correct transfer answers. Token KL has lower evaluation MSE than score fitting and yields zero net-correct transfer change. Average score fidelity, teacher agreement, and gold correctness give different comparisons.

For score fitting, the transfer rescue/harm pair is 14/11. Its descriptive one-sided exact paired-binomial tail is 0.345019. Conditional on the 736 originally correct Q answers, the one-sided 95% Clopper–Pearson upper limit for harm is 2.461732%. The fixed reference required both evaluation MSE reductions to exceed or equal 10%, no increase in teacher disagreements, and transfer net gain at least 11 with nominal $p \leq 0.05$ and harm upper limit at most 2%. It additionally compared score fitting with each distillation objective on fidelity, disagreements, and net gain. The combined reference is unmet. Because these panels had already been examined by the project, the inferential quantities are reported as descriptive diagnostics.

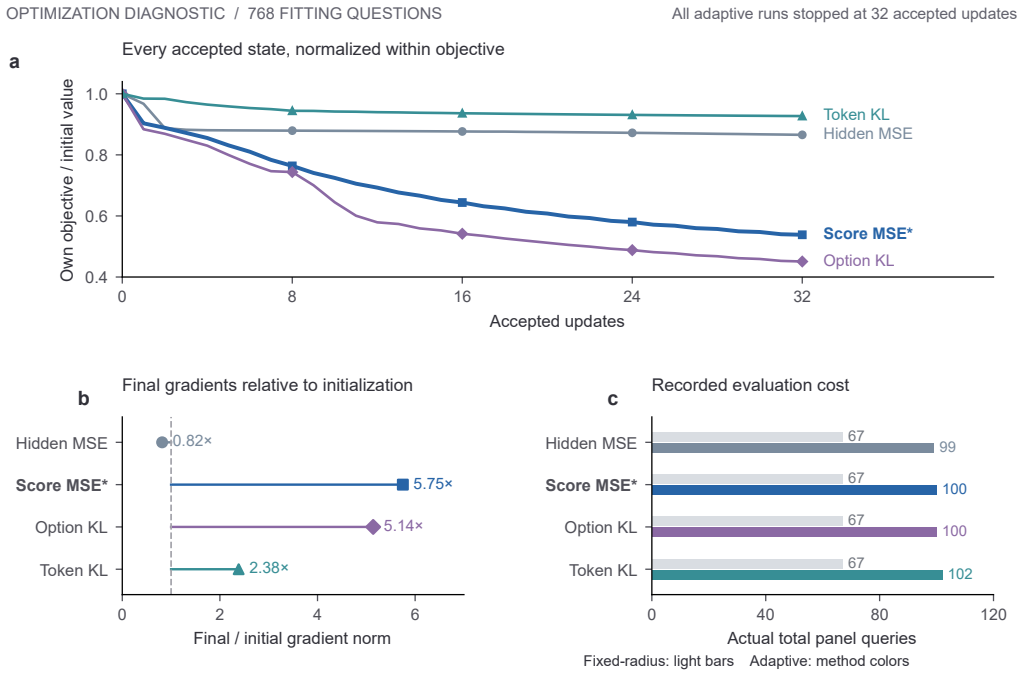

Figure 10: Optimization progress and recorded evaluation cost. **a**, Each arm's fitting objective at every accepted state, normalized to its initial value. **b**, Final-to-initial joint gradient norm; the dashed line marks one. **c**, Actual total head-evaluation panels for fixed-radius (light) and adaptive (coloured) runs, including initial/final diagnostics and transfer. The asterisk denotes the designated score objective. All adaptive arms reach 32 accepted updates. Each panel evaluates both student and teacher; a fresh full-model check is counted separately. Curves describe one trajectory per objective.

Table 8: Actual adaptive optimization and resource accounting. Total panels include initial, final-gradient, and transfer evaluations; each panel evaluates both student and teacher heads. Every arm also performs one fresh full-model check.

Objective	Own loss ↓ (%)	$\|g_f\|/\|g_0\|$	Opt. panels	Total panels	Time (s)
Hidden MSE	13.44	0.8165	96	99	1130.03
Score MSE	46.17	5.7484	97	100	1105.67
Option KL	54.92	5.1404	97	100	1112.59
Token KL	7.29	2.3847	99	102	1107.36

All final losses are lower than in the fixed-radius comparison, while all runs terminate at the 32-acceptance cap rather than the gradient threshold. Each arm records 33 gradient panels, including a gradient recomputed at the final accepted state. For hidden MSE, the matched 768-question regularized reference $dL_{\text{hidden}} + \lambda\|AB\|_F^2$ is 107.941193 with $\lambda = 0.0002639841856168936$, compared with the analytical ridge minimum 96.243164. Its gap decreases from 13.769762 in the fixed-radius diagnostic to 11.698028. This regularized reference differs from the optimized unregularized hidden loss.

Computation and verification. Each worker uses one GPU and two CPU threads. Final-score reanalysis reproduces rescue/harm and MSE; full-model checks match 16,384 scores exactly. Cross-platform counts and decisions agree, with floating differences at most 4.51×10^{-17}. The supplement reanalyses saved outputs on CPU. Full gradient tensors remain archived; fresh computation requires the model assets and execution environment.