Greedy Decoding Is Not Precision-Invariant: Cross-Precision Output Divergence in LLM Inference*

Gaoyuan Du *gaoyuan.du@utk.edu*
University of Tennessee, Knoxville

Anam Nawaz Khan *akhan59@utk.edu*
University of Tennessee, Knoxville

Rex Zhou *rexzhe1230@gmail.com*
University of Chicago

Xiaoyang Liu *lxaoya@amazon.com*
Amazon

Deepayan Chakrabarti *deepay@utexas.edu*
University of Texas at Austin

Fnu Suya *fsuya@utk.edu*
University of Tennessee, Knoxville

Xueping Li *Xueping.Li@utk.edu*
University of Tennessee, Knoxville

Reviewed on OpenReview: *https://openreview.net/forum?id=QDOKyg7a5e*

Abstract

Greedy decoding from large language models is commonly treated as deterministic. We show it is not *precision-invariant*: the same model, prompt, and decoding algorithm produce different outputs in BF16 versus FP16 on identical hardware. Across our evaluations of six models (1.1B–7B parameters, four families; divergence additionally characterised at 12B) and three benchmarks, 49–100% of prompts diverge; a single token flip often cascades into trajectory-level divergence. We develop an empirical error-propagation analysis and find that 22 layers of accumulated body error do not distinguish flipping from non-flipping steps; the outcome depends primarily on the top-two logit margin at the lm_head relative to the directional perturbation between the top-two candidates. The analysis makes five testable predictions about intervention outcomes, including that applying *more* FP32 compute (broader scope) makes agreement *worse*. The experiments match all five predictions. The best-performing low-overhead intervention we evaluate, selective FP32 lm_head recomputation, triggered only when the margin falls below a threshold, delivers +22–36pp exact agreement on A10G (+12–21pp on L4 and A100) at $< 4\%$ latency overhead in low-batch (batch size ≤ 4) single-stream inference. We map the applicability boundary across six models and four batch sizes, and hypothesise that training-time precision stability is a determining factor. The method is a partial mitigation rather than a universal determinism guarantee: its benefit vanishes when body-originated error dominates, including at batch size ≥ 8 and under end-to-end FP8 in our tests.

*An LLM-based writing assistant was used for language editing and formatting checks. All scientific claims and analyses were verified by the authors.

1 Introduction

Precision switching in LLM serving is a routine operational reality, not a hypothetical concern. Model hubs distribute a single checkpoint that serving frameworks cast to the deployment format; the same model may be served in BF16 on one replica and FP16 on another depending on library defaults, configuration flags, or operator choices. The same model, prompt, and greedy-decoding configuration can therefore produce *different* outputs depending on which numerical format is active. Under our controlled setup, standard reproducibility tooling (`torch.use_deterministic_algorithms`, fixed CUBLAS workspace) makes repeated runs bit-identical within each precision but does not reconcile outputs across precisions. A system can thus be repeatable within each format while failing cross-format replay. In principle, greedy decoding is deterministic: at each step t, the model selects $y_t = \arg\max_v z_t(v)$.

Why does cross-format agreement matter? First, *reproducibility and auditing*: when exact replay is required, an audit replica running a different numerical format may fail to reproduce the logged token sequence. Second, *evaluation validity*: benchmark scores and A/B tests can conflate model quality with format-induced output drift; two evaluations of the *same* checkpoint can disagree on individual predictions while reporting similar aggregate accuracy. Third, *fidelity to a higher-precision reference*: we show (Appendix XIII) that our intervention moves the BF16 arm closer to FP32 greedy inference (+19–21pp sequence agreement). FP16 alone remains closer to FP32 than BF16+C is, so where the serving format is a free choice, FP16 is the better single-format proxy; Intervention C addresses the case where the format is *not* a free choice. The relevant pair of formats will change as hardware evolves, but the mechanism—a low top-two margin meeting format-dependent rounding at the lm_head—can be tested across formats. With FP8 confined to the head projection, a format-rescaled gate achieves exact agreement on TinyLlama and +56pp on Qwen2.5-3B. This controlled result is not representative of end-to-end FP8: when every linear layer is quantised, divergence becomes body-dominated and the head-scope repair recovers only +1pp (Appendix XV).

Running the same model with the same prompt under BF16 vs. FP16 produces different outputs on 59–70% of TinyLlama-1.1B prompts across public benchmarks (GSM8K, HumanEval, MBPP). The rate rises to 82% on Qwen2.5-3B-Instruct and reaches 100% on the tested BF16-saturated Qwen variants. The cross-model intervention table covers six models from 1.1B to 7B across four families (Llama, Qwen, Mistral, and OLMoE), and a separate 12.2B experiment characterises divergence without running the full intervention comparison.[1]

The mechanism is more specific: hidden-state error accumulates *uniformly* through the transformer body (present in every prompt regardless of outcome), and a token flips when some competitor's differential perturbation exceeds its original logit gap; empirically, such flips concentrate at steps with a small top-two margin at the lm_head (Section 7.5). This observation motivates the remedy: selectively recompute the lm_head in FP32 at steps where the margin falls below a gating threshold τ.

This paper makes four contributions:

1. **A systematic characterisation of cross-precision output divergence.** We measure the phenomenon primarily at the *output* level across six models (1.1–7B, four families), four benchmarks, and three GPU microarchitectures: 49–100% of greedy generations differ between BF16 and FP16, with a mean length difference of 34 tokens, and on Qwen2.5-3B GSM8K 19% of prompts flip final-answer correctness. Numerical differences across formats are well known; our measurements quantify their output-level extent and show how one token flip can cascade through an autoregressive trajectory.
2. **A predictive mechanism for cross-precision greedy divergence.** A token flips between BF16 and FP16 when some competitor's differential perturbation exceeds its original logit gap. Empirically, flips concentrate where the top-two margin is small relative to the per-logit perturbation scale ($\sigma_z \approx 0.026$). Four independent experiments support this characterisation—divergence-conditioned tracing, layer grafting, KV-cache grafting, and logit-lens traceback—all converging on the lm_head

[1]The six models in Table 12 are TinyLlama-1.1B-Chat, Llama-3.2-3B-Instruct, Qwen2.5-3B-Instruct, Mistral-7B-Instruct-v0.3, OLMoE-1B-7B, and DS-R1-Distill-Qwen-7B. The 12.2B probe uses Mistral-Nemo-Instruct-2407. Two additional Qwen variants (Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct) exhibit BF16 saturation that precludes a meaningful intervention comparison.

as the dominant amplification stage. The directional projection of body error onto the top-two decision direction supports lm_head dominance (Section XII). The mechanism replicates across three tasks and two model families.

3. **Five controlled interventions test the mechanism's predictions.** The mechanism predicts each intervention's outcome *before* the experiment is run: integer quantization and temperature sharpening have zero effect; top-K FP32 recomputation works for all $K \geq 2$; full FP32 lm_head recomputation matches top-K; and extending FP32 scope beyond the lm_head degrades agreement (63%→44%). All five predictions are confirmed.

4. **An applicability map across models, kernels, and batch sizes.** We evaluate the intervention on six models (1.1–7B, four families), additionally characterise divergence at 12B, test two attention kernels, and vary batch size (bs) over $\{1, 2, 4, 8\}$ (with composition experiments extending to bs= 16). The lift ranges from +36pp (Llama-3.2-3B) to 0pp (Qwen BF16-saturated); effectiveness appears to track training-time precision stability (hypothesis). Two boundaries emerge: the remedy survives FlashAttention-2 in the tested setting, but the lm_head-only scope vanishes at bs$\geq$8.[2]

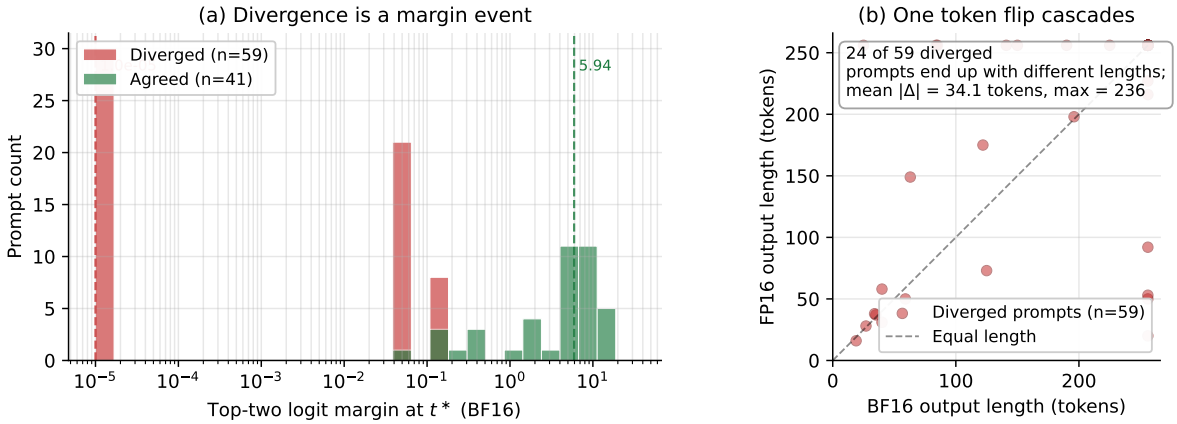

Figure 1: Two empirical facts that shape this paper (TinyLlama-1.1B, GSM8K, BF16 vs. FP16 greedy decoding, $n = 100$). **(a)** Distribution of the top-two logit margin at the measurement step t^*, for prompts whose outputs diverge (red) and for matched agreed controls (green). The two groups are bimodal across more than five orders of magnitude. DIVERGED median $\approx 10^{-5}$, with 30/59 exactly at 0 (perfect BF16 ties); AGREED median ≈ 5.9. Divergence is a *margin event* at the lm_head, not an error-magnitude event in the transformer body. **(b)** Sequence-length scatter of BF16 vs. FP16 outputs on the 59 DIVERGED prompts. A single token flip typically cascades into trajectory-level divergence: 24/59 prompts end up with different lengths, with a mean length difference of 34 tokens and maximum difference of 236 tokens.

2 Related work

As machine learning moves into audited deployments, reproducibility can be a deployment requirement in its own right, not merely a byproduct of accuracy. Floating-point non-determinism in training is classical (Goldberg, 1991; Higham, 2002; Zhuang et al., 2022). Framework-level determinism tools (PyTorch Team, 2020) are effective within a fixed format but have no effect on cross-precision divergence (Section 6.1).

[2]Throughout this paper, "Pass@1" denotes greedy-decoding final-answer correctness on a benchmark—e.g. on GSM8K, the extracted final number—and is computed under our bare-prompt format with no system prompt or chat template, then McNemar-tested in matched-pair form between Baseline and Intervention C. For HumanEval, we additionally report execution-based pass@1 (Appendix XVI): execution-outcome agreement far exceeds token-level agreement (100% vs. 32% on TinyLlama-1.1B; 97.6% vs. 26.2% on Qwen2.5-3B-Instruct, which solves half the benchmark), and Intervention C introduces no observed pass@1 degradation on either model. Pass@1 numbers therefore lower-bound what each model can achieve under its native chat template (Appendix VI, the chat-template results).

BF16/FP16 serving is standard (Dettmers et al., 2022; Micikevicius et al., 2018; Kalamkar et al., 2019), but its output-level consequences and targeted mitigation remain incompletely characterised.

The closest prior work is LayerCast (Yuan et al., 2025), which stores weights in BF16 and casts weight-relevant computation to FP32 layer by layer to reduce system-level numerical variation. LayerCast also observes a connection between small top-token gaps and numerical output changes; we build on that observation by quantifying the margin's role through grafting and gating experiments, localising the dominant correctable component to the lm_head, and exploiting it for a selective head-scoped intervention. We do not implement LayerCast. Our separate global FP32-compute baseline upcasts the BF16/FP16-stored model before inference and adds +38% latency on A10G, whereas the selective head-scoped intervention adds $< 4\%$. PrecisionDiff (Wang et al., 2026) also studies BF16/FP16 output disagreements, systematically identifying *where* precision sensitivity appears across model components. PrecisionDiff characterises the sensitivity surface, while we analyse the decision-level mechanism, quantify each stage's contribution via grafting interventions, and derive a selective repair guided by an exact directional flip condition. Concurrent work (He et al., 2025; Atil et al., 2024; Qi et al., 2025; Gond et al., 2026; Fu et al., 2026) targets kernel- or batch-level sources orthogonal to format-level divergence, or characterises nondeterminism without localising its mechanism. Messina & Scotta (2026) formalise "background temperature" to characterise the effective stochasticity of floating-point perturbations at $T{=}0$; our work complements that framework with a mechanistic diagnosis and a targeted fix. To our knowledge, prior work has not combined output-level characterisation, stage-wise causal interventions, and selective lm_head recomputation. Low-precision formats are known to produce numerical differences; we study their *extent* at the output level (49–100% of greedy generations diverge in our evaluations), their *mechanism* (late low-margin events with an lm_head-dominated correctable component), and their *partial controllability* (a margin-gated intervention recovering 22–36pp of exact agreement on A10G). Prior work on numerical (non)determinism also addresses within-format effects from batching and kernel scheduling (He et al., 2025; Atil et al., 2024), which are distinct from the cross-format comparison studied here.

3 Problem formulation

Consider an autoregressive LM with vocabulary V. At step t, the model computes logits $z_t = f_\theta(x, y_{<t})$ and selects $y_t = \arg\max_v z_t(v)$ (Vaswani et al., 2017). In floating-point arithmetic, z_t depends on the numerical precision (BF16, FP16, or FP32), tensor-parallelism degree, and GPU architecture.

We define the *top-two logit gap* $\Delta_t^{(c)} = z_t^{(c)}(v^{(1)}) - z_t^{(c)}(v^{(2)})$ under configuration c. When this gap is small enough that floating-point rounding across configurations can flip the argmax, different configurations select different tokens. The autoregressive conditioning then diverges, producing a cascade of length $L_{\text{cascade}} = T - t_{\text{div}}$.

We measure four quantities. The **Exact Agreement Rate** (EAR) is the fraction of prompts producing identical token sequences across configurations. The **First Divergence Step** (t^*) is the position of the first token mismatch. The **Trigger Rate** is the fraction of steps where $\Delta_t < \tau$ (used only when a selective intervention applies). The **Mean Gap** ($\bar{\Delta}$) averages Δ_t over all decoding steps. For tasks with well-defined final answers, we additionally report **Semantic Agreement Rate** (SAR)—whether the extracted conclusion matches across configurations even if the full reasoning differs.

4 Inference-time interventions for cross-precision agreement

The contribution of this paper is a mechanistic diagnosis of cross-precision divergence—it is a top-two margin event localised at the lm_head—together with a controlled ablation that rules out alternative explanations. We study three inference-time interventions within this ablation: each one modifies only the final decoding step (without retraining, modifying the transformer body, or adding persistent GPU memory) but tests a different hypothesis about the mechanism. Intervention A asks whether the issue is comparison-level (a tie-breaking artefact between BF16 and FP16 logit values that happen to be representable differently). Intervention B asks whether a small-K FP32 correction is sufficient, and is a direct empirical probe of the rank-1-vs-rank-2 prediction. Intervention C asks whether full FP32 lm_head recomputation eliminates

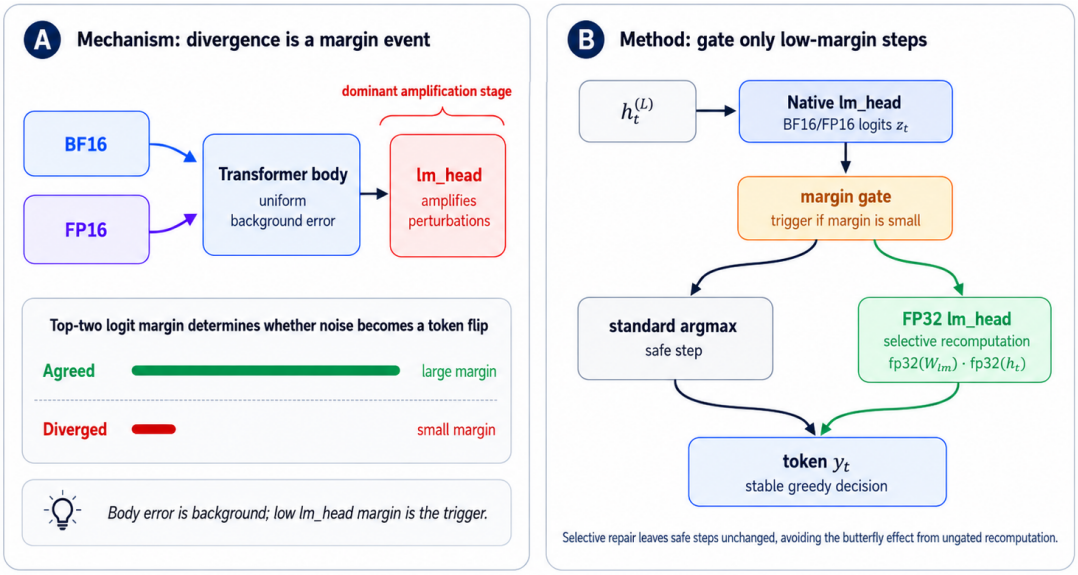

Figure 2: Mechanism and method. **(A)** Cross-precision divergence is a *margin event*: BF16 and FP16 accumulate uniform background error through the transformer body, and the lm_head—the dominant amplification stage—turns that error into a token flip only when the top-two logit margin is small. Prompts with a large margin agree across precisions; prompts with a small margin diverge. **(B)** Our method gates on this margin: the native (BF16/FP16) lm_head logits z_t are computed first, and only when the margin is small does the gate trigger a selective FP32 lm_head recomputation ($\text{fp32}(W_{\text{lm}}) \cdot \text{fp32}(h_t)$); large-margin (safe) steps take the standard argmax unchanged. Selective repair ($\sim$1.4% of steps) leaves safe steps untouched, avoiding the butterfly effect of ungated recomputation.

divergence, and is also the most natural deployable fix. The three interventions provide a sequence of mechanism tests: success of A would support a comparison-level explanation; if A fails but B with $K = 2$ succeeds, this supports a rank-1-vs-rank-2 correction mechanism in the tested gated setting; if B at $K = 2$ matches C at $K = |V|$, recomputing beyond the top two provides no additional benefit in this experiment, supporting an lm_head-localised correction with a 4000$\times$ smaller intervention. Figure 2 illustrates the shared selective-gating envelope; the three interventions below differ only in what they manipulate inside the gated step.

Predictions from the analysis. An error-propagation analysis (Section 7, $\sigma_z \approx 0.026$) predicts: A has zero effect, B and C produce comparable lifts (rank-1-vs-rank-2 only), and extending scope beyond lm_head hurts. All predictions confirmed.

Selective triggering. All interventions fire only when $\Delta_t < \tau = 10^{-3}$ (0.7–1.4% of steps), keeping overhead $<4\%$.

4.1 Intervention A: Integer logit quantization

The argmax over floating-point logits can differ across precisions because BF16 and FP16 represent the same real number with different roundings; integer comparison after quantization is bit-exact. Intervention A exploits this by quantizing logits into coarse bins of width τ before argmax, testing whether divergence is comparison-level:

$$\tilde{z}_t(v) = \text{round}\left(\frac{z_t(v)}{\tau}\right), \quad y_t = \arg\max_v \tilde{z}_t(v). \tag{1}$$

Table 1: Inference-time interventions for cross-precision agreement. Cost is per-trigger overhead relative to a standard decoding step.

Intervention	Approach	Per-Trigger Cost	Mechanism
A. Integer quantization	Reduce precision	$O(\|V\|)$ div+round	Merge small-gap logits
B. Top-K FP32 + int compare	Increase then reduce	$O(K \cdot d)$ FP32 matmul	FP32 recompute + int merge
C. Full-vocab FP32 lm_head	Increase precision	$O(d \cdot \|V\|)$ FP32 matmul	FP32 recompute of lm_head

Any two logit values within τ of each other map to the same integer. Ties in the integer domain are broken by selecting the smaller token index, ensuring a canonical, platform-independent decision. This is a *lossy* approach that deliberately discards sub-τ differences. The cost is negligible: one division and one round per vocabulary element.

4.2 Intervention B: Top-K FP32 recomputation with integer comparison

Intervention A assumes that cross-precision logits are close enough to fall within the same quantization bin. When they are not—i.e., the logit *values* themselves differ across precisions—the quantization step alone cannot reconcile them. Intervention B addresses this by first correcting the logit values through FP32 recomputation of the top-K candidates, then applying integer quantization for a deterministic final comparison:

1. Extract the top-K candidate indices $\mathcal{K} = \text{top-}K(z_t)$ from the native-precision logits.
2. Recompute the lm_head in FP32 for only those K rows: $\hat{z}_t(\mathcal{K}) = \text{float32}(W_{\text{lm}}[\mathcal{K}]) \cdot \text{float32}(h_t^{(L)})$, where $W_{\text{lm}}[\mathcal{K}] \in \mathbb{R}^{K \times d}$ is the sub-matrix of lm_head weights corresponding to the K candidates.
3. Quantize: $\tilde{z}_t(v) = \text{round}(\hat{z}_t(v)/\tau)$ for $v \in \mathcal{K}$, then integer argmax with canonical tie-break (smallest token index wins).

This is a *hybrid* approach that combines value correction (partial FP32 recompute) with comparison stabilization (integer binning). The cost per trigger is one partial FP32 matrix multiply of size $K \times d$ ($K = 8$, $d = 2048$: 16K FLOPs), plus negligible quantization overhead—4000× cheaper than Intervention C's full $|V| \times d$ recompute per trigger. We use $K = 8$ as a conservative default; Appendix IX shows that $K = 2$ suffices (all values $K \in \{2, 4, 8, 16, 32, |V|\}$ produce identical EAR), showing that, for the tested gated cases, correcting the top-two candidates is sufficient to match the full-vocabulary version.

4.3 Intervention C: Full-vocabulary FP32 lm_head recomputation

The dominant source of cross-precision logit perturbation is the lm_head: a single matrix multiplication $z_t = W_{\text{lm}} \cdot h_t^{(L)}$ projecting the d-dimensional hidden state onto the full vocabulary $|V|$. Because $|V|$ is large (32K–152K), this operation amplifies small hidden-state differences by a factor of $\|W_{\text{lm}}\|$. Intervention C eliminates this amplification by recomputing the projection in FP32 over the full vocabulary:

$$\hat{z}_t = \text{float32}(W_{\text{lm}}) \cdot \text{float32}(h_t^{(L)}), \tag{2}$$

where both activations and weights are up-cast to FP32 on-the-fly, keeping GPU memory constant (no persistent FP32 weight copies). The recomputation scope can in principle extend to the last L transformer layers, trading additional compute for deeper error correction, though our scope ablation (Section 7.6) shows this *degrades* agreement in practice. This is the most expensive intervention per trigger but directly targets the dominant error source identified by our layer-level analysis (Section 7.5). The cost is one full FP32 matrix multiply over the entire vocabulary ($d \times |V|$).

4.4 Overhead and trade-off analysis

Table 1 summarizes the three interventions.

With empirical trigger rates of 0.7–1.4%, all interventions add <4% latency under selective gating. The key methodological question is whether cross-precision divergence is caused by *small* logit differences (within τ,

Algorithm 1 Inference-time intervention for cross-precision agreement

Require: Model f_θ, prompt x, threshold τ, max tokens T, intervention $\in$ {A, B, C}

for $t = 1$ **to** T **do**
  $z_t \leftarrow f_\theta(x, y_{<t})$; $v^{(1)}, v^{(2)} \leftarrow$ top-2(z_t); $\Delta_t \leftarrow z_t(v^{(1)}) - z_t(v^{(2)})$
  if $\Delta_t < \tau$ **or** $\tau = 0$ **then**
    $y_t \leftarrow \text{Apply}(f_\theta, x, y_{<t}, z_t, \text{A/B/C})$
  else
    $y_t \leftarrow v^{(1)}$
  end if
  if $y_t =$ `<eos>` **then**
    break
  end if
end for

addressable by integer quantization) or *large* differences (ranking reversals, requiring FP32 recomputation). Our experiments answer this definitively.

We additionally evaluate two supplementary baselines in Appendix IX: **(D) Temperature sharpening** (multiply logits by $\alpha > 1$ before argmax, zero cost) and **(E) Consensus decoding** (run BF16 and FP16 in parallel, FP32 tiebreak on disagreement, $\sim 2\times$ cost). Neither changes the cost–agreement frontier established by A–C: sharpening has zero effect (confirming the value-level nature of divergence), and consensus achieves perfect agreement only at double the compute budget.

5 Experimental setup

Public benchmarks. Our primary evaluation uses three public benchmarks: GSM8K (Cobbe et al., 2021) (math, 8-shot CoT), HumanEval (Chen et al., 2021) (code, 0-shot), and MBPP (Austin et al., 2021) (code, 3-shot).

Models. Our primary experiments (Sections 6.2–6.3) use TinyLlama-1.1B-Chat (Zhang et al., 2024) as the main testbed due to its fast iteration time and ability to run full ablations on a single GPU. For model sensitivity analysis (Section 6.5), we compare reasoning-tuned models (DeepSeek-R1-Distill-Qwen-1.5B/7B (DeepSeek-AI, 2025)) against instruction-tuned models (Qwen2.5-1.5B/7B-Instruct (Qwen Team, 2024)).

Configuration. We compare BF16 vs. FP16 greedy decoding on the same GPU (A10G 22GB). Max generation: 256 tokens. Fixed seeds throughout. To isolate numerical-format effects from PyTorch-level kernel non-determinism, we also run a sanity-check ablation (Section 6.1) that varies `torch.use_deterministic_algorithms` across {False, True} with `CUBLAS_WORKSPACE_CONFIG=:4096:8` and `cudnn.benchmark=False`, confirming that under greedy decoding with batch size 1 and fixed seed, PyTorch kernels are already self-deterministic within a single precision and the flag has no effect on cross-precision divergence.

Why BF16 vs. FP16. We focus on BF16 vs. FP16 because both are common 16-bit inference formats on modern GPUs (Dettmers et al., 2022; Kwon et al., 2023; NVIDIA, 2023), yet they allocate exponent and mantissa bits differently (8/7 versus 5/10). This pair provides a controlled test of representation-dependent output divergence. We directly test transfer to head-isolated FP8/BF16 and to FP16/FP32 on T4; applying the methodology to other format pairs remains future work.

Baselines. Greedy BF16/FP16 (standard serving precision), Greedy FP32 (oracle upper bound—eliminates all precision-induced divergence), global FP32 compute (upcast the BF16/FP16-stored model before inference), and ungated FP32 recomputation (Intervention C with $\tau = 0$, recomputing every step without selective gating).

Table 2: BF16 vs. FP16 divergence spectrum (TinyLlama-1.1B, $n = 100$). Exact agreement (EAR) ranges from 30% to 41%. 95% CI: Wilson score interval.

Dataset	Domain	EAR (%)	95% CI
MBPP	Code	30.0	[21.9, 39.6]
HumanEval	Code	36.0	[27.3, 45.8]
GSM8K	Math	41.0	[31.9, 50.8]

6 Results

Unless otherwise noted, all results in this section use TinyLlama-1.1B-Chat with BF16 vs. FP16 greedy decoding ($n = 100$ per dataset). Additional results with DeepSeek-R1 and Qwen2.5 models appear in Section 6.5.

6.1 Sanity check: kernel-level determinism is not the cause

Before attributing divergence to numerical format, we rule out two alternative explanations:

(1) PyTorch kernel non-determinism. Toggling `torch.use_deterministic_algorithms`, `CUBLAS_WORKSPACE_CONFIG`, and `cudnn.benchmark` between "loose" and "strict" settings in independent subprocesses produces *bit-identical* outputs within each precision: BF16 self-consistency is 100% and the 41% BF16-vs-FP16 EAR persists identically across both arms. The pattern generalises to MoE (OLMoE-1B-7B) and batch sizes $\{1, 4, 8, 16, 32\}$ (Appendix VIII).

(2) GEMM kernel selection difference. cuBLAS may select different GEMM algorithms for BF16 vs. FP16 matmuls. Two observations rule this out. First, BF16 self-consistency is 100%, so kernel choice is deterministic within a dtype. Second, running the same BF16 weights under native BF16 vs. FP32 compute (different kernels, same weights) yields 42% EAR, indistinguishable from the 41% baseline where both dtype and kernel differ. The divergence is caused by BF16's arithmetic rounding, not by kernel dispatch.

6.2 Divergence is substantial across tasks

Table 2 summarizes BF16-vs-FP16 divergence across all datasets.

On code generation tasks, 64–70% of prompts produce different outputs between BF16 and FP16. On math reasoning, 59% diverge.

Where does divergence happen? The first divergence step t^* has a median of 34–50 tokens on non-Qwen models (the divergence-step analysis), with 44% diverging in the first 30 tokens and a long tail past token 200. Qwen2.5-7B-Instruct is catastrophically early-biased (median $t^* = 1$, 100% first-token divergence), consistent with training-time BF16 saturation (Section 6.5). The left-skewed distribution motivates per-step gating: most divergences are localised to a small fraction of steps identifiable by the top-two margin.

6.3 Intervention comparison: a controlled ablation

Table 3 presents the central experiment: all three interventions and baselines on BF16 vs. FP16 ($\tau = 10^{-3}$).

Five findings emerge, building from root cause to practical recommendation:

(1) The FP32 oracle confirms the root cause. Running both configurations in FP32 yields 100% agreement: all divergence originates from precision differences, not from non-determinism in the decoding algorithm.

(2) Global FP32 compute achieves high agreement but at high cost. Global FP32 upcasting reaches 88–89% agreement—the highest among all methods short of the FP32 oracle—but doubles memory footprint,

Table 3: Intervention and baseline comparison: BF16 vs. FP16 exact agreement (%) (TinyLlama-1.1B, $n = 100$, $\tau = 10^{-3}$). Gated FP32 lm_head recomputation (B, C) achieves the best efficiency–agreement trade-off among the tested methods. All per-cell 95% Wilson CIs have half-width ±9–10pp at $n = 100$; because Baseline and Intervention C are evaluated on the *same* prompts, the relevant tests are the matched-pair McNemar ($p < 0.001$) and the paired bootstrap CI on the lift ($[+11, +33]$pp, Appendix X) rather than the per-cell intervals. Full CIs for each cell are reported in Appendix X. Appendix X replicates the Baseline / Intervention C rows at $n = 300$ (GSM8K), full-benchmark HumanEval ($n = 164$), and full-split MBPP ($n = 257$) to confirm the lift survives at larger sample sizes (all $n = 100$ cells sit inside the scaled CIs). Appendix X additionally reports 10 000-sample bootstrap confidence intervals for every headline lift in the paper; the TinyLlama +22pp EAR gain has a 90% bootstrap CI of $[+11, +33]$pp.

Method	GSM8K	HumanEval	MBPP	Overhead
Baseline (no intervention)	41.0	36.0	30.0	—
Baselines				
Greedy FP32 (oracle)	100.0	100.0	100.0	∼2× mem
Global FP32 compute	88.0	88.0	89.0	∼2× mem
Ungated FP32 recompute ($\tau = 0$)	60.0	56.0	54.0	∼2.5×
Gated interventions ($\tau = 10^{-3}$)				
A. Integer quantization	41.0	36.0	30.0	<1%
B. Top-K FP32 + integer compare	**63.0**	**61.0**	**55.0**	<2%
C. Full-vocab FP32 lm_head	**63.0**	**61.0**	**55.0**	<4%

preventing deployment on the same hardware for larger models. Gated FP32 lm_head recomputation (Interventions B/C) achieves lower absolute agreement (55–63%) but with <4% overhead, making it practical for production serving.

(3) Ungated recomputation (τ=0) underperforms gating. Recomputing at every step achieves only 54–60% agreement at ∼2.5× latency, *worse* than gated Interventions B/C (55–63% at <4% overhead). Global recomputation changes the logit landscape at safe steps, introducing new divergence points downstream (Section 7).

(4) Integer quantization has zero effect. Intervention A produces identical agreement to the baseline across all datasets, confirming that cross-precision divergence is caused by *value-level* logit differences (BF16 and FP16 compute different numbers), not *comparison-level* ties (same numbers compared differently by argmax).

(5) Gated FP32 recomputation is the best trade-off. Interventions B and C achieve identical improvements (+22pp on GSM8K, +25pp on HumanEval and MBPP) with <4% overhead, outperforming all baselines on the efficiency–agreement frontier. Gated FP32 lm_head recomputation also raises SAR from 58% to 72% on GSM8K. The <4% overhead is backed by direct measurement on a single A10G (Appendix VIII). Gated C adds +1.4% latency and +11% peak memory versus the BF16 baseline; the global FP32-compute baseline adds +38% latency under the same measurement.

Where does the +22pp come from? Table 4 stratifies the 100 GSM8K prompts into four mutually exclusive categories based on whether gated C triggers and whether the flip reproduces on single-step re-run. Gated C fires on exactly 30 prompts (those with $\Delta_t < \tau = 10^{-3}$ at the lm_head), and empirically fixes 22 of them (73% success rate on triggered prompts). The 29 prompts in the "margin $\geq \tau$" category are where the flip is driven by accumulated upstream error—gated C correctly declines to trigger there, because a single-step FP32 recomputation cannot reconcile two hidden states that have already drifted apart. This stratification predicts the exact headline lift (41% → 63%) without tuning and explains both where the method applies and where it does not.

Robustness checks. We characterise divergence across six models spanning four families (Llama, Qwen, Mistral, and OLMoE-MoE; 1.1B–7B), with an additional divergence-only probe at 12B. Intervention lifts are

Table 4: Per-prompt stratification of when gated Intervention C applies (TinyLlama-1.1B, GSM8K, $n=100$, $\tau = 10^{-3}$). Categories are mutually exclusive. The main-text +22pp EAR improvement (Table 3) arises because C fires on 30 prompts (B+C) and successfully converts 22 of them ($22/30 \approx 73\%$ of triggered prompts). Category D (29 prompts) is where the flip is driven by large-error accumulation and the lm_head margin is already decisive at t^*; gated C correctly does not trigger there.

Prompt category	Count	Distribution by t^* (first divergence step)					
		$t^*=0$	1–10	11–50	51–100	>100	—
AGREED (no intervention needed)	41				—		
Gated, single-step flip reproducible (C most effective)	25	4	4	10	5	2	—
Gated, flip KV-cache-dependent (C may not fix)	5	—	2	1	1	1	—
Margin $\geq \tau$ (C does not trigger)	29	—	4	8	9	8	—

Table 5: TOST non-inferiority analysis (GSM8K, $\delta = 0.05$, n per row; bare "Solve step by step" prompt format, hence absolute accuracies are lower than the chat-template numbers of Table 15 and are comparable only within this table). We test H_0: IntC accuracy $\leq$ Baseline accuracy $- \delta$. Rejection ($p < 0.05$) establishes that IntC is no worse than 5pp below baseline. Interpretation note: TinyLlama's baseline accuracy (2.3%) is itself below the $\delta = 0.05$ margin, so a 5pp shortfall is arithmetically impossible there and its rejection should be read as a consistency check rather than as evidence; Qwen2.5-3B (11.7%) is the informative row. Paired McNemar on the same Qwen run is non-significant ($\chi^2 = 0.14$, $p = 0.708$; $b = 34$ discordant pairs favouring baseline, $c = 30$ favouring Intervention C). Qwen2.5-3B was re-run at n=600 to raise statistical power: non-inferiority is established ($p = 0.009$), with the accuracy difference's 95% CI $[-4.3, +2.9]$pp lying entirely inside the ±5pp margin.

Model	n	Acc (Baseline)	Acc (IntC)	Diff	TOST p	Non-inferior?
TinyLlama-1.1B	300	2.3%	2.0%	−0.3pp	4.3×10^{-5}	**Yes**
Qwen2.5-3B	600	11.7%	11.0%	−0.7pp	**0.009**	**Yes**

+36pp on Llama-3.2-3B, +8–10pp on Mistral-7B and OLMoE, +3pp on Qwen2.5-3B, and 0pp on the BF16-saturated Qwen variants. Effectiveness appears to track training-time precision stability (hypothesis; not directly validated), rather than scale alone. The phenomenon and intervention persist with FlashAttention-2 and through 1024-token reasoning chains on MATH-500. We find no statistically significant pass@1 degradation in the matched-pair tests; a TOST analysis with margin $\delta = 0.05$ establishes non-inferiority on Qwen2.5-3B at $n = 600$ ($p = 0.009$), while the TinyLlama result is a low-accuracy consistency check rather than informative quality evidence. C's lift degrades from +17pp at bs=1 to 0pp at bs=8; composition with global FP32 compute recovers +5pp at bs=8. Digit tokens are 8× over-represented among flipped tokens.

Counter-intuitive predictions confirmed by the mechanism. Body-layer error does not discriminate between flipping and non-flipping prompts: the 22-layer TinyLlama and the 36-layer Qwen2.5-3B have high divergence rates (59% and 82%) despite different body-error budgets. In these measurements, the lm_head margin distribution is more predictive of a flip than layer count or body-error magnitude.

6.4 Semantic agreement and gating sensitivity

Two findings extend the headline picture along orthogonal axes (full data in appendix).

Semantic exceeds exact agreement. On GSM8K, 58% of prompts produce the same final numerical answer despite different reasoning chains (SAR), compared to 41% exact token match (EAR); Intervention C lifts SAR to 72%. Token differences also need not imply functional differences on code: on Qwen2.5-3B HumanEval, execution outcomes agree on 97.6% of problems while full token sequences agree on only 26.2% (Appendix XVI). Exact agreement therefore measures a stricter property than task-outcome agreement.

Downstream evaluation impact. Cross-precision divergence can be hidden by aggregate benchmark accuracy. On Qwen2.5-3B-Instruct (GSM8K, $n = 100$), 19% of prompts *flip correctness* between BF16 and FP16: 9 prompts are correct only in BF16 and 10 only in FP16, while aggregate accuracy differs by only 1pp (27% vs. 28%). On Llama-3.2-3B-Instruct with the native chat template (66% accuracy), correctness flips drop to 2%. These two cases show that similar aggregate scores can mask different per-prompt outcomes; determining how this effect scales with model accuracy requires broader evaluation.

Selective triggering is essential. On DeepSeek-R1-1.5B ($n = 200$), gated C ($\tau \in [10^{-4}, 10^{-2}]$) achieves +6.5pp with $< 4\%$ overhead, while ungated recomputation ($\tau = 0$) achieves only +2.5pp with 2.5× latency—the "butterfly effect" makes blanket recomputation counterproductive (mechanism explained in Section 7). The threshold is insensitive across two orders of magnitude, consistent with the bimodal margin distribution.

6.5 Model sensitivity: reasoning vs. instruction tuning

At matched parameter scales, reasoning-tuned models show 10–15pp lower exact agreement and 1.7× higher trigger rates than instruction-tuned counterparts (DeepSeek-R1 vs. Qwen2.5; Appendix IV). Chain-of-thought reasoning produces more uncertain intermediate steps, increasing exposure to low-margin events. Longer CoT trajectories accumulate more close-to-tipping-point decoding decisions, each precision-sensitive. This matches the scale analysis accompanying Proposition 1: the divergence rate scales with the lower-tail mass of the margin distribution, and CoT concentrates more of this distribution at small values.

7 Mechanism: an empirical analysis of error propagation

We separate the empirical observation from the analysis. *The observation* is that divergence concentrates at steps with a near-zero top-two margin, while the body's hidden-state error — which is present in every prompt and grows monotonically through the layers — does not discriminate between flipping and non-flipping steps once projected onto the decision direction. *The analysis* builds on that observation: an exact algebraic condition for when a flip occurs, and an estimate of when a head-scope repair can undo one. This section provides an empirical analysis rather than formal bounds on floating-point error propagation. We give one exact algebraic condition (which is elementary), then two quantitative estimates whose assumptions we state and which we evaluate against measurements. The model is organised around one exact result and one criterion: Proposition 1 states when a token flips, and the cross-arm hidden-state gap Δh (Table 6) states when a head-scope repair can undo it. Between them they cover the four format pairs we measure — BF16/FP16, FP16/FP32, head-isolated FP8, and end-to-end FP8 — including the one in which the intervention fails. Two further failure modes, batched decode and cross-kernel variation, are governed by different mechanisms and we treat them separately. We state one exact result (Proposition 1, with proof), develop two quantitative heuristic estimates validated against measurement, and identify the irreducible weight-truncation floor that bounds any inference-time fix. Full derivations are in Appendix I.

7.1 Error propagation: from machine epsilon to logit perturbation

Consider a transformer with L layers and hidden dimension d, producing logits $z_t = W_{\text{lm}} \cdot h_t^{(L)}$ where $W_{\text{lm}} \in \mathbb{R}^{|V| \times d}$. Under precision $c \in \{\text{BF16}, \text{FP16}\}$, each layer introduces a rounding error bounded by $\|\delta_\ell\| \leq \epsilon_c \cdot C_\ell \cdot \|h_\ell\|$, where $\epsilon_{\text{BF16}} \approx 3.9 \times 10^{-3}$ and $\epsilon_{\text{FP16}} \approx 4.9 \times 10^{-4}$ (an 8× gap due to the 7-bit vs. 10-bit mantissa). With residual connections keeping per-layer Jacobian norms $\|J_\ell\| \approx 1$, a worst-case (no-cancellation) accumulation gives $\|\hat{h}_L - h_L\| \leq L \cdot \epsilon_c \cdot \bar{C} \cdot \|h\|$.

Empirically we measure a ∼5× amplification from pre-lm_head ℓ_2 (0.92) to logit ℓ_2 (4.63) on TinyLlama-1.1B. The per-logit perturbation is $\|\hat{z} - z\|/\sqrt{|V|} \approx 4.63/\sqrt{32000} \approx 0.026$. This yields the central divergence condition:

Proposition 1 (Exact directional flip condition) *Fix a decoding step and let $z, \hat{z} \in \mathbb{R}^{|V|}$ be the logit vectors computed by two numerical configurations. Assume each vector has a unique maximiser, let $v^{(1)} = \arg\max_v z(v)$, and write $\Delta z(v) = \hat{z}(v) - z(v)$. Then $\arg\max_v \hat{z}(v) \neq v^{(1)}$ if and only if there exists $v \neq v^{(1)}$*

such that

$$\Delta z(v) - \Delta z(v^{(1)}) > z(v^{(1)}) - z(v). \tag{3}$$

For the original runner-up $v^{(2)}$, this inequality is exactly the condition for $v^{(2)}$ to overtake $v^{(1)}$. It is necessary for a flip whose new winner is $v^{(2)}$; together with $\hat{z}(v^{(2)}) > \max_{u \notin \{v^{(1)}, v^{(2)}\}} \hat{z}(u)$, it is sufficient. Exact ties are resolved by the implementation's deterministic tie-breaking rule.

Proof. Under the uniqueness assumption, $\arg\max_v \hat{z}(v) \neq v^{(1)}$ iff some $v \neq v^{(1)}$ satisfies $\hat{z}(v) > \hat{z}(v^{(1)})$, i.e. $z(v) + \Delta z(v) > z(v^{(1)}) + \Delta z(v^{(1)})$, which rearranges to the stated condition. Taking $v = v^{(2)}$ gives the pairwise crossing condition; checking $v^{(2)}$ against all remaining coordinates identifies it as the new winner. □

On the status of this proposition. The proposition is an algebraic rearrangement, stated separately to distinguish the proved condition from the estimates that follow. The two quantitative accounts below—the perturbation scale and the criterion for when a head-scope repair can undo a flip—are estimates with explicitly stated assumptions.

Scale analysis (heuristic, not part of the proposition). The proposition is exact but involves the unobservable per-coordinate perturbations $\Delta z(v)$. To obtain a practical trigger statistic we approximate their typical magnitude by the vocabulary-wide RMS $\sigma_z = \|\hat{z} - z\|/\sqrt{|V|} \approx 0.026$ (TinyLlama-1.1B), giving the directional difference a scale of $\approx \sqrt{2}\,\sigma_z \approx 0.037$. This motivates the margin-gated trigger of Intervention C: flips concentrate where Δ_t is small relative to this scale. The approximation is deliberately heuristic—per-coordinate perturbations are neither Gaussian nor independent—and we assess its adequacy purely empirically: the directional criterion classifies flip-vs-no-flip correctly on 88–89% of steps across two model families (Section XII), and DIVERGED prompts show median margin 0.0 vs. 2.69 for AGREED (260× mean gap). An alternative sufficient-condition formulation via an ℓ_∞ bound is also available—no flip can occur if $2\|\hat{z} - z\|_\infty < \Delta_t$, since $|\Delta z(v^{(2)}) - \Delta z(v^{(1)})| \leq 2\|\hat{z} - z\|_\infty$—and Intervention C's gate can be read as enforcing this safe condition at a per-format calibrated scale; we use the RMS form in the main text because $\|\hat{z} - z\|_\infty$ is dominated by outlier coordinates that rarely coincide with the top-two candidates, so the ℓ_∞ bound is safe but looser than the RMS-scale heuristic as a trigger statistic.

Empirical validation. Body-layer L2 is indistinguishable between DIVERGED and AGREED groups while the top-two margin separates them by orders of magnitude (Table 7), confirming that the margin—not the error magnitude—determines divergence.

7.2 Why FP32 recomputation works

During autoregressive generation with a KV cache, each decoding step processes only the *new* token through the transformer body; the prefix hidden states are retrieved from cache without recomputation. The fresh rounding error introduced at step t is therefore of order $\epsilon_c \cdot C_{\text{step}}$—dominated by the single lm_head matmul, not the full L-layer body.

Why FP32 recomputation reconciles the paths. The starting point is an immediate observation, not a theorem: given the same hidden state $h_t^{(L)}$, the deterministic FP32 matmul $\text{float32}(W_{\text{lm}}) \cdot \text{float32}(h_t^{(L)})$ produces bit-identical outputs regardless of whether the upstream path was BF16 or FP16. We keep this observation explicit precisely *because* it is trivial: it is what makes the intervention's reach predictable. Since the FP32 recomputation is deterministic, the two arms agree at a gated step *if and only if* the hidden states they feed it agree. At a gated step, therefore, the intervention's reach is governed by the cross-arm hidden-state gap Δh rather than by any property of the head. We use this as a screening quantity, not as a law: it orders the four format pairs we test into regimes (Table 6), but it does not cover batching or kernel variation, and we have tested no configuration in the wide gap between the working and failing regimes. The substantive, empirically testable claim is quantitative: at gated steps the cross-path logit difference should drop from $\sigma_z \approx 0.026$ to the *body-originated residual* only, which the error model of Section 7.3 heuristically

estimates as

$$\sigma_z^{(\text{after C})} \approx \underbrace{\frac{L \cdot \alpha}{\sqrt{d}} \cdot \sigma_z}_{\text{body residual } (\sim 2.4\% \text{ of } \sigma_z)} \approx 6 \times 10^{-4}. \tag{4}$$

For prompts whose divergence is entirely lm_head-originated (body hidden states are effectively identical between precisions), this residual vanishes and C achieves exact reconciliation; the achievable EAR lift then equals the fraction of such prompts. This quantitative account is validated by the reconciliation data: the +22pp improvement matches the fraction of prompts whose divergence originates from single-step margin events—30/100 prompts enter the gate (category B of Table 4), of which 22 are successfully reconciled (73% success rate within scope).

7.3 When lm_head-only scope is sufficient, and when it is not

The analysis above shows C *works*; we now show why the lm_head is the dominant error source—making body-layer FP32 unnecessary.

Heuristic estimate: why the lm_head dominates. Under three explicit assumptions—(i) per-layer Jacobian norms ≈ 1 maintained by residual connections, (ii) rounding errors accumulate without systematic cancellation across layers (conservative, worst-case linear in L), and (iii) the d per-element rounding errors of the lm_head dot product accumulate as $\sqrt{d}$ (standard random-sign model)—the body-to-lm_head perturbation ratio is approximately $L\alpha/\sqrt{d}$, where $\alpha = \|F_\ell(h)\|/\|h\| \ll 1$ is the residual ratio:

$$\frac{\sigma_z^{(\text{body})}}{\sigma_z^{(\text{lm_head})}} \approx \frac{L \cdot \alpha}{\sqrt{d}}. \tag{5}$$

Argument. In KV-cached decoding at step $t > 0$, only the new token passes through L body layers. The residual connection $h^{(\ell)} = h^{(\ell-1)} + F_\ell(h^{(\ell-1)})$ confines rounding error to the F_ℓ branch, whose output norm is $\alpha \cdot \|h\|$ with $\alpha \ll 1$ (a necessary condition for training stability). We bound the body contribution conservatively (worst-case linear accumulation): after L layers the body error is at most $L \cdot \alpha \cdot \epsilon_c \cdot \|h\|$. By contrast, the lm_head computes a d-dimensional dot product on $h_t^{(L)}$ at its full norm; the d rounding errors accumulate to a typical magnitude $\approx \sqrt{d} \cdot \epsilon_c \cdot \|h\|$ (a random-sign estimate, *not* a bound; the worst case is $d \cdot \epsilon_c \cdot \|h\|$, attained only if every rounding error shares a sign). This ratio deliberately combines a worst-case numerator with a typical-case denominator. Under the random-sign assumption, this choice biases the diagnostic toward assigning greater relative importance to body error. Because the denominator remains an estimate rather than a bound, however, the ratio is not a formal bound on either contribution. Adversarial weight configurations in which the d products share a sign would make the head term $\sqrt{d}$ times larger and the ratio correspondingly smaller. We therefore report the ratio only as an order-of-magnitude estimate and assess it empirically: it matches the measured decomposition ($\sim$97.6% head-originated in the storage-vs-computation ablation) and the ordering of intervention lifts across the four precision configurations in Table 6. The actual body accumulation may be sub-linear (statistical cancellation across layers would give $\sqrt{L}$), which would reduce the estimated body-to-head ratio further under the same assumptions. The resulting estimate is $L \cdot \alpha/\sqrt{d}$.

Numerical evaluation. For TinyLlama-1.1B ($L=22$, $d=2048$, $\alpha \approx 0.05$ measured as the mean per-layer ratio $\|F_\ell(h)\|/\|h\|$ across 22 layers and 100 GSM8K prompts via the unconditional layer trace in Appendix II):

$$\frac{22 \times 0.05}{\sqrt{2048}} = \frac{1.1}{45.3} \approx 0.024. \tag{6}$$

Under this heuristic, the body contributes $\sim$2.4% of the per-logit perturbation and the lm_head contributes $\sim$97.6%. FP32 recomputation of the lm_head therefore targets the estimated dominant source, consistent with the +22pp lift. Extending the FP32 scope to body layers addresses only the estimated residual while introducing the butterfly effect (Section 7.6), which is why scope extension empirically *hurts* (Appendix IX: RMSNorm+lm_head drops EAR from 63% to 44% on TinyLlama; Appendix IX replicates the result on DeepSeek-R1-7B with $\tau=0$). Full derivation and step-0 (prefill) edge case in Appendix I.

A conditional form that covers every precision pair we test. The ratio above is derived for two paths that run the *same* body in *different* low precisions, which is the BF16-vs-FP16 setting. Our FP8 experiments (Appendix XV) sit outside that assumption, so we state the estimate in the conditional form that governs all three configurations. The relevant quantity is not the head's share of σ_z but the *cross-arm hidden-state gap* $\Delta h = \|h^{(A)} - h^{(B)}\|/\|h\|$, because Intervention C works by making the two arms compute *identical* logits: at a gated step both evaluate $\mathrm{float32}(W_{\mathrm{lm}}) \cdot \mathrm{float32}(h)$, which coincide only to the extent that h coincides. Writing n_q for the number of quantized linear operations per layer and ϵ_A, ϵ_B for the two arms' body precisions,

$$\Delta h \;\approx\; \max(n_q, 1)\, L\, \alpha\, |\epsilon_A - \epsilon_B|, \tag{7}$$

and C can reconcile a gated step only when Δh is small relative to the margin scale σ_z. This yields an ordering that matches every configuration we measure:

Table 6: The conditional form of the scope estimate, evaluated for all four *format* configurations we test (the batched-decode and cross-kernel boundaries are governed by a different mechanism; see text) (TinyLlama-1.1B: L=22, d=2048, α≈0.05, n_q=7 linear ops per layer). Δh is the cross-arm hidden-state gap from Eq. 7. The ratio column uses a single reference scale, the BF16-vs-FP16 value $\sigma_z \approx 0.026$, so that the four rows are comparable on one axis; the format's own σ_z is larger for FP8 (Appendix XV) and smaller for FP16-vs-FP32, and using per-format values would compress the spread without changing which regime each row falls in.

Configuration	Body across arms	Δh	$\Delta h/\sigma_z$	Predicted / observed
FP8 at head only	bit-identical	0	0	complete / +70pp, EAR 100%
FP16 vs. FP32 (T4)	same body, two precisions	0.0005	∼0.02	head-dominated / +7pp
BF16 vs. FP16 (main)	same body, two precisions	0.004	∼0.14	head-dominated / +22pp
End-to-end FP8 vs. BF16	body quantized in one arm	0.45	∼17	body-dominated / +1pp

The four configurations span nearly three orders of magnitude in $\Delta h/\sigma_z$, and the *regime* tracks that ordering: complete reconciliation when the body is shared ($\Delta h = 0$), a positive lift whenever the body gap stays well below the margin scale ($\Delta h/\sigma_z \leq 0.14$), and a negligible one once it exceeds that scale (17). The raw lift is not monotone in Δh and we do not claim it is: FP16-vs-FP32 yields +7pp at $\Delta h/\sigma_z = 0.02$ against a baseline already at 88%, i.e. it closes 58% of the available headroom, whereas BF16-vs-FP16 yields +22pp from a 41% baseline, closing 37%. What Δh orders is which regime a configuration falls in, not the size of the lift within a regime. The criterion also explains why the two pairs with the *largest* raw precision gaps behave oppositely: FP16-vs-FP32 has a large ϵ ratio but a tiny Δh because FP32 carries the untruncated weights, whereas end-to-end FP8 has both a large ϵ and n_q=7 quantized operations per layer. The $L\alpha/\sqrt{d} \approx 0.024$ figure of the previous paragraph is the special case n_q=1, ϵ_A/ϵ_B = BF16/FP16; it should not be read as a constant. This gives an operational test for a new *format pair*: measure Δh between the two serving configurations on a handful of prompts and compare it to the margin scale. Measured on GSM8K (n=50, final-token hidden state), $\Delta h = 0.0105$ for TinyLlama BF16-vs-FP16 and 0.0219 for Qwen2.5-3B under the same pair — about twice the body gap. We note two limits on reading that comparison causally. The predicted value from Eq. 7 for the TinyLlama pair is 0.004, so the estimate is a factor 2.6 below measurement. And Qwen's lift under this same pair and the same Δh is +3pp on A10G but +14pp on A100 and +20pp on L4 (Appendix XIV), so Δh cannot be what sets the lift's magnitude for that model; the kernel path matters at least as much. We are explicit about the criterion's scope. It applies to *format* pairs evaluated at a fixed batch size, and it does not explain the batched-decode boundary: measuring bs=1 against bs=8 in the same precision gives $\Delta h = 0.0127$, statistically indistinguishable from the working BF16-vs-FP16 configuration, yet the lift there degrades to zero. Batching fails for a different reason, which we treat separately (Section 7.4): a changed reduction order produces a *third* trajectory rather than a larger gap between two, so gated recomputation introduces new divergence instead of removing it. The criterion likewise takes the kernel implementation as fixed — the +3/+14/+20pp spread for Qwen across A10G, A100 and L4 (Appendix XIV) shows the effective ϵ is itself implementation-dependent, a dimension the estimate does not cover.

Table 7: Divergence-conditioned trace (TinyLlama-1.1B, $n = 100$).

	DIVERGED	AGREED	Gap
Body L2 (layer 21)	1.022	1.135	−0.113
Logit L2	4.928	5.559	−0.631
Top-two margin	**0.039**	**5.733**	**−5.694**
Flip rate	58%	0%	—

7.4 Why ungated recomputation hurts

Ungated recomputation ($\tau = 0$) produces a *third* trajectory at safe steps, introducing new divergence downstream ("butterfly effect"). Result: +2.5pp vs. +6.5pp for gated on DeepSeek-R1-1.5B. Gating is mechanistically necessary.

7.5 Empirical validation: layer-level divergence tracing

We trace BF16-vs-FP16 hidden-state L2 through all 22 layers on $n = 100$ GSM8K prompts, comparing DIVERGED (n=59) and AGREED (n=41) at their respective measurement steps. Key finding (Table 7): **body-layer L2 is indistinguishable between groups**, but the **top-two margin differs by** 150× (0.039 vs. 5.733). Four independent causal experiments (layer grafting, KV-cache grafting, logit-lens traceback, Qwen2.5-3B replication; Appendices II–II) confirm: the flip is produced at the lm_head.

7.6 Summary

The analyses above converge on a three-factor causal picture. Cross-precision greedy divergence requires the simultaneous presence of three conditions—remove any one and the outputs agree:

1. **Low arithmetic precision at the lm_head.** BF16's 7-bit mantissa introduces per-element rounding of magnitude $\epsilon_{\text{BF16}} \approx 3.9 \times 10^{-3}$, 8× larger than FP16's 4.9×10^{-4}, so the two precisions compute materially different logits at every step (per-logit perturbation $\sigma_z \approx 0.026$; see Section 7.5).
2. **The lm_head amplifies hidden-state error.** The $d \times |V|$ projection amplifies body-layer ℓ_2 error by ∼5× (0.92 → 4.63); without this amplification, per-logit perturbations would sit at ∼0.005, below typical margins.
3. **Greedy decoding has zero tolerance for rank flips.** Unlike sampling, greedy argmax converts any logit-ranking reversal into a hard token difference, and a single flipped token cascades through autoregressive conditioning (mean length difference 34 tokens, Figure 1).

Intervention C breaks condition (1) at the dominant correctable stage. Recomputing the lm_head in FP32 ($\epsilon_{\text{FP32}} \approx 6 \times 10^{-8}$) reduces the head-originated perturbation from ∼0.03 to $\sim 10^{-5}$. Conditions (2) and (3) remain unchanged: the lm_head still amplifies its input and greedy decoding still selects the argmax, but the corrected head contributes much less arithmetic error. The storage-vs-computation ablation supports this decomposition quantitatively: upgrading only the matmul compute path to FP32 (while keeping storage at BF16) drives EAR from 0.41 to 0.82, and the corresponding FP32-compute / FP32-compute pair sits at 0.88 rather than 1.00, indicating that ∼87% of the reachable EAR gap is BF16-arithmetic-origin (addressable by C) while a residual ∼12pp is associated with weight truncation. In the evaluated ablations, extending the FP32 scope beyond lm_head (to RMSNorm or the last transformer layer) reduces EAR from 63% to 44–52% (Appendix IX); this shows that broader recomputation is not automatically better and can introduce new downstream trajectory differences.

8 Discussion

8.1 Practical implications

Intervention C is intended for low-batch greedy decoding (bs$\leq$4), a regime that can arise in code completion, structured-output generation, benchmark evaluation, and audit-oriented replay. In our A10G measurements it is a low-overhead intervention ($< 4\%$; Section 6.3), but its benefit should be measured on the target model and platform. Two scope boundaries apply: (i) models with FP16-body NaNs or body-dominated divergence require a body-scope or training-time remedy, and (ii) at bs$\geq$8 the head-only intervention provides no lift by itself in our tests and must be composed with a body-scope method (Appendix VIII). The effect remains positive with both attention kernels tested (eager and FlashAttention-2) and on A10G, L4, and A100, although the lift varies substantially by GPU (Appendix XIV). Precision and kernel configuration should therefore be recorded explicitly rather than assuming platform-independent agreement.

8.2 What does partial agreement provide?

Exact whole-sequence agreement is a strict metric: one differing token anywhere in a 256-token generation counts as disagreement. In the tested BF16/FP16 setting, Intervention C raises this metric from 30–41% to 55–63% at $< 4\%$ A10G latency overhead, but it does not provide bit-exact reproduction. The practical value is application-dependent. On Qwen2.5-3B HumanEval, execution outcomes agree on 97.6% of problems while token sequences agree on 26.2% (4 of 164 execution outcomes flip; Appendix XVI); on GSM8K, however, Qwen2.5-3B shows a 19% cross-precision correctness-flip rate. These results rule out treating either token disagreement or task neutrality as universal proxies for the other. The TOST analysis establishes non-inferiority within a 5pp margin on Qwen2.5-3B ($n = 600$, $p = 0.009$), while TinyLlama's very low baseline accuracy makes its non-inferiority result only a consistency check (Table 5).

The head-isolated FP8 experiment is a controlled upper-bound diagnostic: sharing the BF16 body and weight storage removes body-originated divergence and the BF16/FP16 weight-truncation floor by construction, after which a format-rescaled gate reaches 100% exact agreement on TinyLlama and gains +56pp on Qwen2.5-3B. It is not representative of end-to-end FP8 serving. When every linear layer is quantised to FP8, the same head-scope repair yields only +1pp (Appendix XV). Applications requiring bit-exact replay should pin the full numerical configuration or use an end-to-end higher-precision reference rather than relying on Intervention C alone.

8.3 Practitioner decision procedure

The deployment decision tree (Section 7.6) summarises the action tiers. The screening metric, the coefficient of variation of trigger density (CV(TD)), requires only a single BF16 run ($\sim$5–10 minutes, no FP16 comparison needed). Retrospectively, it separates the models in our study into the observed action tiers; across the five models with a defined lift comparison, CV(TD) is associated with Intervention C's EAR lift (Spearman $\rho = 0.90$, Pearson $r = 0.84$). This is an in-sample screening heuristic, not a validated cross-model predictor, and should be recalibrated on the target hardware.

8.4 Limitations

We characterise divergence across six models from four families at scales from 1.1B to 7B and include a divergence-only probe at 12B; the full intervention comparison is not evaluated at either the 12B or 70B+ scale. Most experiments use an A10G, with selected headline replications on L4 and A100 and an FP16/FP32 probe on T4. The direction of the intervention effect is consistent in these replications, but its magnitude is hardware-dependent. C is evaluated for greedy decoding and is effective by itself only at bs$\leq$ 4 in our batch-size sweep; at bs$\geq$8 and under end-to-end FP8, divergence becomes body-dominated and a head-only repair is insufficient. We do not evaluate sampling-based decoding. C does not guarantee full agreement, and the $\sim$12pp BF16/FP16 weight-storage gap measured in our ablation cannot be removed by changing only the lm_head arithmetic.

9 Conclusion

LLM greedy decoding is not precision-invariant: 49–100% of prompts diverge between BF16 and FP16 across the models and benchmarks we test (59–82% on the four models that are neither BF16-saturated nor at the low end). Divergence concentrates at low top-two margins relative to the directional perturbation, with a 150× gap between the measurement-step margins of divergent and convergent prompts in the headline trace. Gated FP32 lm_head recomputation raises exact agreement by +22–36pp on A10G at <4% latency overhead and is the best-performing low-overhead intervention among those evaluated. Selected L4 and A100 replications yield positive but hardware-dependent lifts of +12–21pp. The intervention is evaluated up to the 7B scale (with divergence additionally characterised at 12B), on dense and MoE models, and is limited to low-batch greedy serving. In the controlled head-isolated FP8 setting, a format-rescaled gate reaches exact agreement on TinyLlama and gains +56pp on Qwen2.5-3B; in end-to-end FP8, where body error dominates, the gain is only +1pp. Overall, the results identify low-margin lm_head events as an important and partially controllable source of cross-precision divergence, while also delineating regimes in which a head-scope intervention is insufficient.

Broader impact

Cross-precision reproducibility can matter when exact replay, audit trails, or per-example evaluation records are required. Our intervention narrows the measured agreement gap at <4% A10G latency overhead in low-batch (bs≤4) greedy inference and requires no model retraining, but it does not guarantee identical outputs. We see no direct misuse pathway introduced by the method: it changes numerical consistency rather than model capability. Reproducibility is not correctness; a repeatable but wrong model remains wrong, and this work is not a substitute for accuracy, safety, or domain-specific validation.

References

Berk Atil, Sarp Aykent, Alexa Chittams, et al. Non-determinism of "deterministic" LLM settings. *arXiv preprint arXiv:2408.04667*, 2024.

Jacob Austin, Augustus Odena, Maxwell Nye, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Ranjith Chodavarapu and Lei Xu. The illusion of equivalence: Systematic FP16 divergence in KV-cached autoregressive inference. *arXiv preprint arXiv:2604.15409*, 2026.

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

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *NeurIPS*, 2022.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *NeurIPS*, 2022.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized language models. In *NeurIPS*, 2023.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *ICLR*, 2023.

Tairan Fu, Gonzalo Martínez, Javier Conde, Carlos Arriaga, Pedro Reviriego, Xiuyuan Qi, and Shanshan Liu. Beyond reproducibility: Token probabilities expose large language model nondeterminism. *arXiv preprint arXiv:2601.06118*, 2026.

Alireza Ghaffari, Justin Yu, Mahsa Ghazvini Nejad, Masoud Asgharian, Boxing Chen, and Vahid Partovi Nia. Mitigating outlier activations in low-precision fine-tuning of language models. *arXiv preprint arXiv:2312.09211*, 2023.

David Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, 23(1):5–48, 1991.

Raja Gond, Aditya K. Kamath, Ramachandran Ramjee, and Ashish Panwar. LLM-42: Enabling determinism in LLM inference with verified speculation. *arXiv preprint arXiv:2601.17768*, 2026.

Horace He and Thinking Machines Lab. Defeating nondeterminism in LLM inference. Thinking Machines Lab technical blog, September 2025. `https://thinkingmachines.ai/blog/defeating-nondeterminism-in-llm-inference/`.

Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 2nd edition, 2002.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *ICLR*, 2020.

Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, et al. A study of BFLOAT16 for deep learning training. *arXiv preprint arXiv:1905.12322*, 2019.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, et al. Efficient memory management for large language model serving with PagedAttention. In *SOSP*, 2023.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *ICML*, 2023.

Yingru Li, Jiawei Xu, Jiacai Liu, Yuxuan Tong, Ziniu Li, Tianle Cai, Ge Zhang, Qian Liu, and Baoxiang Wang. Dynamic vocabulary pruning: Stable LLM-RL by taming the tail. *arXiv preprint arXiv:2512.23087*, 2025.

Ji Lin, Jiaming Tang, Haotian Tang, et al. AWQ: Activation-aware weight quantization for LLM compression and acceleration. In *MLSys*, 2024.

Peter Mattson, Christine Cheng, Cody Coleman, et al. MLPerf training benchmark. In *MLSys*, 2020.

Alberto Messina and Stefano Scotta. Introducing background temperature to characterise hidden randomness in large language models. *Transactions on Machine Learning Research*, 2026.

Paulius Micikevicius, Sharan Narang, Jonah Alben, et al. Mixed precision training. In *ICLR*, 2018.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. OLMoE: Open mixture-of-experts language models. *arXiv preprint arXiv:2409.02060*, 2024.

NVIDIA. TensorRT-LLM: A TensorRT toolbox for optimized large language model inference. *GitHub Repository*, 2023.

Xu Ouyang, Tao Ge, Thomas Hartvigsen, Zhisong Zhang, Haitao Mi, and Dong Yu. Low-bit quantization favors undertrained LLMs: Scaling laws for quantized LLMs with 100T training tokens. *arXiv preprint arXiv:2411.17691*, 2024.

Hung Viet Pham, Shangshu Qian, Jiannan Wang, et al. Problems and opportunities in training deep learning software systems: An analysis of variance. In *ASE*, 2020.

PyTorch Team. Reproducibility: Controlling sources of randomness. `https://pytorch.org/docs/stable/notes/randomness.html`, accessed 2026. Also: `torch.use_deterministic_algorithms` and the `CUBLAS_WORKSPACE_CONFIG` environment variable.

Penghui Qi, Zichen Liu, Xiangxin Zhou, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Defeating the training-inference mismatch via FP16. *arXiv preprint arXiv:2510.26788*, 2025.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

SGLang Team. Towards deterministic inference in SGLang and reproducible RL training. LMSYS technical blog, September 2025. https://www.lmsys.org/blog/2025-09-22-sglang-deterministic/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. In *NeurIPS*, 2017.

Yifei Wang, Tianlin Li, Xiaohan Zhang, Xiaoyu Zhang, Wei Ma, Mingfei Cheng, and Li Pan. Hidden reliability risks in large language models: Systematic identification of precision-induced output disagreements. *arXiv preprint arXiv:2604.19790*, 2026.

Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

Guangxuan Xiao, Ji Lin, Mickael Seznec, et al. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *ICML*, 2023.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, et al. ZeroQuant: Efficient and affordable post-training quantization for large-scale transformers. In *NeurIPS*, 2022.

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. In *NeurIPS (Oral)*, 2025.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. TinyLlama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

Ziyang Zhang, Xinheng Ding, Jiayi Yuan, Rixin Liu, Huizi Mao, Jiarong Xing, and Zirui Liu. Deterministic inference across tensor parallel sizes that eliminates training-inference mismatch. *arXiv preprint arXiv:2511.17826*, 2025.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, et al. SGLang: Efficient execution of structured language model programs. In *NeurIPS*, 2024.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics (TACL)*, 2024.

Juntang Zhuang, Tommy Tang, Yifan Ding, et al. Randomness in neural network training: Characterizing the impact of tooling. In *MLSys*, 2022.

Appendix overview

The supplementary material is organised into six groups, each self-contained:

- **Analysis details** (App. I): the proof context for the exact flip condition and full derivations of the heuristic error-propagation estimates.
- **Mechanism evidence** (App. II–II): four causal experiments isolating divergence to the lm_head.
- **Cross-model robustness** (App. III–VII): replication across six models, four families, math and code benchmarks.
- **Production-stack robustness** (App. VIII): FlashAttention, batch sizes, global-FP32 composition, overhead.
- **Ablations** (App. IX): scope, threshold, top-K, alternative interventions.
- **Statistical validation and reproducibility** (App. X–XI): CIs, scaled replication, code.

Cross-references between appendix sections are kept to a minimum.

I Error-propagation analysis: derivations and assumptions

This appendix presents a heuristic error-propagation model. With the exception of Proposition 1 (exact), the estimates here rest on the stated simplifying assumptions and are validated empirically rather than proven; counterexamples to the asymptotic rates can be constructed under adversarial weight configurations. The model predicts the empirical findings in the main text and characterises the scope boundary of Intervention C.

I.1 Setup and notation

Let f_θ denote an L-layer autoregressive transformer with hidden dimension d and vocabulary size $|V|$. At decoding step t, the model computes:

$$h_t^{(\ell)} = \text{Layer}_\ell(h_t^{(\ell-1)}), \quad \ell = 1, \ldots, L, \tag{8}$$

$$z_t = W_{\text{lm}} \cdot h_t^{(L)}, \quad W_{\text{lm}} \in \mathbb{R}^{|V| \times d}, \tag{9}$$

$$y_t = \arg\max_v z_t(v). \tag{10}$$

Under floating-point precision c with machine epsilon ϵ_c, each arithmetic operation has relative error bounded by ϵ_c (Goldberg, 1991; Higham, 2002). The relevant values are $\epsilon_{\text{BF16}} = 2^{-8} \approx 3.9 \times 10^{-3}$ (7-bit mantissa) and $\epsilon_{\text{FP16}} = 2^{-11} \approx 4.9 \times 10^{-4}$ (10-bit mantissa), giving a precision ratio of $\epsilon_{\text{BF16}}/\epsilon_{\text{FP16}} = 8$.

I.2 Error accumulation through the transformer body

Heuristic estimate: body-layer error accumulation. Let $\hat{h}_t^{(\ell)}$ denote the hidden state computed under precision c (with the other precision as reference). Under three explicit assumptions—(i) standard transformer architecture with residual connections $h^{(\ell)} = h^{(\ell-1)} + F_\ell(h^{(\ell-1)})$, (ii) per-layer relative rounding error bounded by $\epsilon_c C_\ell$ with condition number C_ℓ of order unity, and (iii) no systematic error cancellation across layers (worst case)—the accumulated error satisfies:

$$\|\hat{h}_t^{(L)} - h_t^{(L)}\| \leq \epsilon_c \cdot \|h\|_{\text{rms}} \cdot \sum_{\ell=1}^{L} C_\ell \prod_{j=\ell+1}^{L} (1 + \epsilon_c C_j). \tag{11}$$

When $\epsilon_c C_\ell \ll 1$ for all ℓ (satisfied for both BF16 and FP16 on standard transformers where C_ℓ is of order unity), this simplifies to linear accumulation:

$$\|\hat{h}_t^{(L)} - h_t^{(L)}\| \approx \epsilon_c \cdot \bar{C} \cdot L \cdot \|h\|_{\text{rms}}, \tag{12}$$

where $\bar{C} = (1/L)\sum_\ell C_\ell$ is the mean per-layer condition number. This is a growth-rate estimate under the stated assumptions, not a theorem about arbitrary weight configurations; its adequacy is assessed empirically below.

Empirical validation. On TinyLlama-1.1B ($L = 22$, $d = 2048$), we measure BF16-vs-FP16 hidden-state L2 growing linearly from 0.002 at layer 0 to 0.92 at layer L (Appendix II), consistent with the linear-accumulation regime.

I.3 Logit perturbation and the divergence condition

Derivation of the heuristic scale analysis. Proposition 1 gives the exact flip condition: the argmax changes if and only if there exists a competitor whose differential perturbation exceeds its original logit gap to the top-1 token. For the original runner-up, this reduces to a comparison between its signed directional perturbation difference and the top-two margin $\Delta_t = z_t(v^{(1)}) - z_t(v^{(2)})$. Here we derive the heuristic scale estimate that turns this exact but unobservable condition into a practical trigger statistic. Let $\sigma_z = \|\hat{z}_t - z_t\|/\sqrt{|V|}$ denote the RMS per-logit perturbation. Under the simplifying assumption that perturbations spread approximately uniformly across the $|V|$ logit dimensions, propagating the body-error estimate above through the lm_head gives:

$$\sigma_z \approx \frac{\|W_{\text{lm}}\| \cdot \epsilon_c \cdot \bar{C} \cdot L \cdot \|h\|_{\text{rms}}}{\sqrt{|V|}}, \tag{13}$$

and a flip becomes plausible only when $\Delta_t \lesssim \sigma_z$, so that heuristically

$$P(\text{flip at } t) \lesssim P(\Delta_t < \sigma_z), \tag{14}$$

which depends primarily on the *margin distribution* of the model at that step, not on the *magnitude* of the accumulated body error. This is an approximation, not a bound: per-coordinate perturbations are neither Gaussian nor independent, and its adequacy is assessed empirically.

Empirical validation. For TinyLlama-1.1B: $\sigma_z \approx 4.63/\sqrt{32000} \approx 0.026$. Measured margins: DIVERGED median = 0.039 (just above σ_z), AGREED median = 5.733 (220× above σ_z). The scale analysis correctly predicts: (i) body-layer L2 does not distinguish the two groups (confirmed: Table 7); (ii) the margin is the dominant factor (confirmed: 150× gap); (iii) the K-ablation result ($K = 2$ suffices, Appendix IX), because a perturbation of magnitude $\sigma_z \approx 0.026$ cannot bridge a rank-2-to-rank-3 gap that is typically $\gg 0.1$.

I.4 Intervention C: error reduction at the lm_head

Quantitative account: why Intervention C reconciles the paths. FP32 lm_head recomputation eliminates the cross-path logit difference at the lm_head by making both precision paths compute identical FP32 logits from their respective hidden states—an immediate consequence of the determinism of the FP32 matmul, not a theorem. For prompts where the BF16 and FP16 hidden states $h_t^{(L)}$ are effectively identical (the "lm_head-originated" regime), both paths produce bit-identical FP32 logits and no flip can occur. Under the same simplifying assumptions as above, the *lm_head-specific* rounding contribution is heuristically reduced by a factor of $\epsilon_{\text{FP32}}/\epsilon_{\text{BF16}} \approx 1.5 \times 10^{-5}$:

$$\sigma_z^{(\text{lm_head, FP32})} = \frac{\epsilon_{\text{FP32}}}{\epsilon_{\text{BF16}}} \cdot \sigma_z^{(\text{lm_head, BF16})} \approx 0.026 \times 0.976 \times 1.5 \times 10^{-5} \approx 4 \times 10^{-7}. \tag{15}$$

The residual cross-path difference after C is the body-originated component only (∼2.4% of the original σ_z, per the dominance estimate of Section 7.3). For lm_head-originated divergences where body hidden states are shared, this residual vanishes and:

$$P(\text{flip at } t \mid \text{FP32 lm_head, lm_head-originated}) = 0. \tag{16}$$

The achievable EAR lift then equals the fraction of prompts whose divergence is entirely lm_head-originated:

$$\text{EAR}_{\text{max}}^{(C)} = \text{EAR}_{\text{baseline}} + \frac{|\{i : \text{prompt } i \text{ diverges only at lm_head}\}|}{n}. \tag{17}$$

Empirical validation. On TinyLlama-1.1B, 30/100 prompts have divergence triggered by a margin event at the lm_head (Table 4, category B+C); of these, 22 are successfully reconciled by C (73% success rate within the scope). The predicted maximum lift is 30%; the achieved lift is 22pp (= 22/100), consistent with a 73% reconciliation rate within the reachable set identified by the analysis.

I.5 Extended derivation and edge cases for the lm_head dominance estimate

The main text (Section 7.3) states and numerically evaluates the lm_head dominance estimate. Here we provide the full derivation and discuss an important edge case.

Full derivation. Decompose the total per-logit perturbation into body and lm_head contributions:

$$\sigma_z = \sigma_z^{(\text{body})} + \sigma_z^{(\text{lm_head})}. \tag{18}$$

Under standard transformer architecture with KV cache, we bound each contribution directly in per-logit units. Consider a single logit position v with row vector $W_v \in \mathbb{R}^d$ ($\|W_v\| \approx 1$ under Xavier scaling):

$$\sigma_z^{(\text{body})} \lesssim \frac{L \cdot \alpha \cdot \epsilon_c \cdot \|h\|}{\sqrt{d}}, \tag{19}$$

$$\sigma_z^{(\text{lm_head})} \approx \epsilon_c \cdot \|h\|. \tag{20}$$

Body term. Each of L layers introduces error only on the non-residual branch F_ℓ (norm $\alpha \cdot \|h\|$), so the accumulated body error is $\|\Delta h_{\text{body}}\| \leq L \cdot \alpha \cdot \epsilon_c \cdot \|h\|$. Projected onto the logit direction W_v, the per-logit effect is $|W_v \cdot \Delta h| \approx \|\Delta h\|/\sqrt{d}$ (standard projection of a d-dimensional vector onto a fixed unit direction).

lm_head term. The dot product $z_v = \sum_j W_{vj} h_j$ accumulates d independent rounding errors. Each multiply-accumulate step contributes an error of order $\epsilon_c \cdot |W_{vj} h_j| \approx \epsilon_c \cdot \|h\|/\sqrt{d}$ (using $|W_{vj}| \sim 1/\sqrt{d}$, $|h_j| \sim \|h\|/\sqrt{d}$). Under standard error-summation arguments with partial sign cancellation, $\sqrt{d}$ such terms accumulate to a typical magnitude $\approx \epsilon_c \cdot \|h\|$. Taking the ratio directly:

$$\frac{\sigma_z^{(\text{body})}}{\sigma_z^{(\text{lm_head})}} \approx \frac{L \cdot \alpha}{\sqrt{d}}, \tag{21}$$

consistent with the main-text dominance estimate of Section 7.3.

Edge case: step $t = 0$ (prefill). At step 0, the model processes the entire prompt through all L layers without a KV cache. Unlike cached decoding, where each step introduces fresh rounding error from only one token's pass through the body, the prefill pass accumulates rounding error across all T_{prompt} token positions in the attention reductions. Each layer's attention output is a weighted sum over T_{prompt} value vectors, and the floating-point non-associativity of this reduction introduces $\approx \sqrt{T_{\text{prompt}}}$ additional error terms per layer under typical-case cancellation. The body-error contribution to the dominance ratio therefore scales roughly as $L \cdot \alpha \cdot \sqrt{T_{\text{prompt}}}/\sqrt{d}$, which can approach or exceed 1 for long prompts ($T_{\text{prompt}} \gtrsim d/(L^2\alpha^2) \approx 850$ for TinyLlama). In this regime, the lm_head is no longer the dominant error source, and Intervention C's lm_head-only scope becomes insufficient—consistent with our observation that first-token divergence on Qwen models is irreducible by any lm_head-scope method (Section 6.5).

I.6 Scope boundary: when Intervention C fails

The error model above also predicts the *failure modes*:

Body-dominated divergence (Tier 3). If a model's training-time precision (typically BF16) produced weights whose FP16 cast yields unstable activations (NaN or extreme values at intermediate layers), the body-layer error $\|\hat{h}_t^{(L)} - h_t^{(L)}\|$ is no longer small relative to $\|h\|_{\text{rms}}$. In this regime, σ_z is dominated by the body contribution rather than the lm_head matmul alone, and FP32 recomputation of the lm_head addresses only a fraction of the total perturbation. This predicts 0pp lift on Qwen BF16-saturated models (confirmed: DS-R1-Qwen-7B, Table 12).

Batch-induced divergence. At batch size ≥ 2, attention reductions over padded sequences introduce an additional source of floating-point variation whose magnitude grows with batch size. This is not simply

a larger cross-arm gap: measured bs=1-vs-bs=8 hidden states differ by $\Delta h = 0.0127$, comparable to the working BF16-vs-FP16 configuration (Section 7.3). The reduction order instead makes the batched arm a *third* trajectory, so an FP32 head recompute reconciles neither pair, and C's lift vanishes (confirmed: 0pp at bs= 8, Appendix VIII).

Weight-truncation floor. A storage-vs-computation ablation shows that even configurations that both compute in FP32 (`store BF16 / compute FP32` vs. `store FP16 / compute FP32`) reach EAR = 0.88 rather than 1.00, leaving a residual ∼12pp gap attributable to $W_{\text{BF16}} \neq W_{\text{FP16}}$ (the same FP32 weights truncated to different mantissa widths). This component of cross-precision divergence is *irreducible* under the two fixed low-precision storage formats—it enters through the weights themselves, not through the computation. Full agreement in this comparison requires a shared storage format, such as loading both arms from the same FP32 weights.

I.7 Intervention comparison from the error model

The error model also explains the other interventions:

Intervention A (integer quantization). Quantization bins have width $\tau = 10^{-3}$. The cross-precision logit difference at divergent steps is $\sigma_z \approx 0.026$—26× larger than the bin width. The two precisions therefore map to *different* bins with probability ≈ 1, and quantization faithfully preserves the disagreement. Prediction: zero effect. Confirmed.

Intervention D (temperature sharpening). Multiplying logits by $\alpha > 1$ is a monotone transformation: $\arg\max_v(\alpha \cdot z_t(v)) = \arg\max_v z_t(v)$ for all $\alpha > 0$. The margin becomes $\alpha \cdot \Delta_t$ and the perturbation becomes $\alpha \cdot \sigma_z$; their ratio Δ_t/σ_z is invariant. Prediction: zero effect at any α. Confirmed (Appendix IX).

Intervention E (consensus decoding). Running both precisions and using FP32 as tiebreaker eliminates all format-level divergence by construction. The cost is 2× compute. The per-step disagreement rate $P(\text{token_BF16} \neq \text{token_FP16})$ at any single step is bounded by $P(\Delta_t < \sigma_z)$; we measure this at 44.5% of steps (Appendix IX), consistent with the heavy left tail of the margin distribution on GSM8K.

Top-K sufficiency. The error model predicts that a token ranked k-th can only be promoted to rank 1 if the perturbation σ_z exceeds the gap between rank 1 and rank k. Since $\sigma_z \approx 0.026$ and the typical rank-2-to-rank-3 gap is $\gg 0.1$ (median ≈ 0.5 on TinyLlama), only rank-2 tokens can realistically be promoted. Prediction: $K = 2$ suffices. Confirmed: all $K \in \{2, 4, 8, 16, 32, |V|\}$ produce identical EAR (Appendix IX).

Group 2 backs the lm_head-localisation claim of Section 7.5 with four mutually independent experiments that triangulate the same conclusion: (i) the unconditional per-layer L2 trace (App. II) characterises the typical hidden-state error budget; (ii) layer grafting (App. II) converts the correlational claim into a causal one; (iii) KV-cache grafting (App. II) isolates the autoregressive-cache contribution; (iv) per-layer logit-lens top-2 traceback (App. II) shows BF16 / FP16 agree on the top-1 token at 97–98% of body layers and disagree only at the lm_head, with cross-task generalisation to HumanEval and MBPP. The four experiments converge on a single causal claim: divergence is produced at the last transformer layer, the final RMSNorm, and the lm_head matmul, the modules Intervention C targets or sits immediately downstream of — C recomputes the lm_head projection only, and works because that is the one stage where an FP32 correction lands on both paths' logits without perturbing upstream state.

II Mechanism evidence: four causal experiments

Summary. (i) Unconditional trace: L2 grows linearly from 0.002 (layer 0) to 0.92 (layer 21), amplified 5× by lm_head to 4.63. **(ii) Layer grafting:** body-layer grafting fixes 36% of flips; Layer 21 + RMSNorm fixes 65–71%. **(iii) KV-cache grafting:** swaps convert 36–48% of flips; partial causal contributor. Consistently, 25/59 DIVERGED prompts agree under single-step re-run (no precision-specific cache) while diverging autoregressively—i.e., ∼42% of flips require the accumulated cache state, while the remaining ∼58% (34/59)

reproduce from current-step arithmetic alone. **(iv) Logit-lens traceback:** BF16/FP16 agree on top-1 at ≥93% of body layers; disagreement is one-shot at lm_head.

The four protocols and their full result tables follow. Together they answer a specific diagnostic question—*does the BF16/FP16 disagreement begin before or inside the lm_head?*—from four independent directions: an unconditional error budget, a causal graft of upstream activations, a causal graft of the autoregressive cache, and a per-layer readout of the decision itself.

(i) Unconditional per-layer hidden-state trace

Protocol. We trace the BF16-vs-FP16 hidden-state ℓ_2 distance through all 22 TinyLlama-1.1B layers on the *first* generated token, averaged over $n = 50$ GSM8K prompts, and separately record the logits after the lm_head. This measurement is *unconditional*: it does not condition on whether the prompt eventually diverges, so it characterises the typical error budget rather than the flip event.

Result. Error originates at layer 0 ($\ell_2 = 0.0021$, relative error 0.39%, cosine similarity 0.999993) and accumulates approximately linearly to $\ell_2 = 0.9341$ at layer 21, with relative error staying below 1.3% at every layer and cosine similarity never dropping below 0.99992. The final RMSNorm (3.4×) and the lm_head (5×) then provide two additional amplification stages, taking the logit-level ℓ_2 to 4.6275. On the matched $n = 100$ summary measurement the same picture holds: $\ell_2 = 0.915$ before the lm_head and 4.641 after (amplification 5.1×), with relative error *falling* from 1.05% to 0.50% and cosine similarity rising from 0.99994 to 0.99999. Crucially, all 100 first-token measurements agree across BF16 and FP16: the first generated token never flips on this model, because the implied per-logit perturbation ($4.63/\sqrt{|V|} \approx 0.026$) sits well below the mean top-two margin at that step ($\bar{\Delta}_t = 0.70$). The error budget is therefore always present and is by itself insufficient to produce a flip—which is what motivates the three conditional and causal experiments below.

(ii) Causal attribution via layer grafting

Setup. The divergence-conditioned trace of Section 7.5 is *correlational*: it shows body-layer L2 is indistinguishable between DIVERGED and AGREED groups, which suggests—but does not prove—that body-layer error is not the causal driver. We convert the claim into a causal one via layer grafting. For each valid DIVERGED prompt where the flip reproduces under single-step re-run ($n = 34/100$), we run a BF16 forward pass at context = prompt $\| y_{<t^*}$ but *override* the output of one module ℓ with the corresponding FP16 hidden state h_ℓ^{FP16}, then allow BF16 kernels to process the grafted activation through all subsequent layers plus the BF16 lm_head. We classify the resulting argmax as "fix (FP16)" if it matches the FP16 target, "keep (BF16)" if it matches the BF16 baseline, or "other".

Findings. Two causal conclusions follow.

(1) Body-layer hidden states are not sufficient to determine the flip. If upstream hidden-state error were the causal driver, replacing it with the FP16 counterpart should fix the flip near 100% of the time. Instead, body-layer grafts fix only ∼36% of flips, and no single body layer is distinguished from the others. Crucially, even the embedding graft—where the *entire* downstream BF16 computation operates on FP16 input tokens—fixes only 29.4% of flips, with 70.6% of prompts still producing the BF16 argmax. This shows that BF16 kernel arithmetic itself acts as an attractor: given any hidden-state input, a BF16 forward pass tends to produce the same argmax as the all-BF16 baseline. The causal driver is therefore the *arithmetic path*, not the input activations.

(2) Causal responsibility concentrates in the last transformer layer and the final RMSNorm. Only grafts at Layer 21 and Final RMSNorm fix flips substantially above the body baseline (65% and 71% vs. ∼36%), a ∼30pp jump that occurs in exactly the last two modules before the lm_head. This localises causal responsibility to the last two computation stages, consistent with the unconditional per-layer error trace above: both stages are the ones whose BF16 arithmetic most directly shapes the lm_head's input. The residual 21% of "keep BF16" outcomes at the RMSNorm graft reflects flips attributable to the BF16

Table 8: Causal attribution via layer grafting (TinyLlama-1.1B, GSM8K, $n = 34$ valid DIVERGED prompts). The FP16 hidden state at each layer is spliced into the BF16 forward pass; downstream BF16 kernels (and the BF16 lm_head) produce the final argmax. Body-layer grafts (Layers 0–20) have a flat ∼36% fix rate, well below 100%, demonstrating that the BF16 kernel path acts as an "attractor" that pulls grafted activations back to the BF16 argmax. The decisive causal jump occurs at Layer 21 (the final transformer layer) and the final RMSNorm, which together fix 65%–71% of flips.

Graft site	Fix (→FP16) %	Keep (→BF16) %	Other %
Embedding	29.4	70.6	0.0
Layer 0	47.1	52.9	0.0
Layer 5	29.4	70.6	0.0
Layer 10	38.2	58.8	2.9
Layer 15	47.1	52.9	0.0
Layer 20	35.3	64.7	0.0
Layer 0–20 (mean)	∼36	∼62	∼2
Layer 21 (last body)	**64.7**	26.5	8.8
Final RMSNorm	**70.6**	20.6	8.8

lm_head matmul itself (the final arithmetic stage that no upstream graft can modify), consistent with the storage-vs-computation decomposition (Section 7.6) into kernel arithmetic and storage effects.

Tables 7 and 8 provide complementary evidence: the divergence-conditioned trace shows body-layer error is *statistically* the same across DIVERGED and AGREED prompts, and the graft experiment shows body-layer hidden states are *causally* not sufficient to determine the flip. The causal weight sits in the last transformer layer, the final RMSNorm, and the lm_head matmul—the three stages that Intervention C's FP32 recomputation (Section 4) already targets or sits immediately downstream of.

(iii) Causal role of the KV cache

Setup. The graft experiment above uses a single-step forward at context = prompt $\| y_{<t^*}$ and therefore eliminates a confound: in the original autoregressive runs, the BF16 and FP16 paths carry *precision-specific* KV caches at step t^*, whereas the single-step re-run recomputes everything from scratch. Indeed, 25/59 DIVERGED prompts show argmax agreement under single-step re-run while still disagreeing under autoregressive greedy—proof that some flips are KV-cache-dependent. We now ask: does the cache itself *cause* these flips, or does it merely amplify upstream error so that the current-step lm_head margin becomes flippable?

To probe this, we re-run each DIVERGED prompt autoregressively to step t^* in each precision, keeping the final `past_key_values`. We then compare four configurations for the single step that emits the next token:

- *Baseline BF16*: BF16 model with BF16 cache → $\arg\max = y^{\text{BF16}}$
- *Baseline FP16*: FP16 model with FP16 cache → $\arg\max = y^{\text{FP16}}$
- *Graft A*: BF16 model with *FP16* cache (cast to BF16 dtype)
- *Graft B*: FP16 model with *BF16* cache (cast to FP16 dtype)

We restrict to prompts where Baselines 1 and 2 disagree ($n = 59$): the cache is then provably the only difference between the two precisions that is carried into the next forward pass.

Findings. Three observations.

(1) The KV cache is a partial, not exclusive, causal source. Graft A converts 47.5% of DIVERGED prompts from BF16 argmax to FP16 argmax, proving that for almost half the flips the precision-specific cache is itself sufficient. For the remaining 52.5%, the BF16 kernel operating on the FP16 cache still produces the

Table 9: KV-cache graft ablation (TinyLlama-1.1B, GSM8K, $n = 59$ DIVERGED prompts). Each row swaps the cache between BF16 and FP16 paths for the single forward step that produces y_{t^*}. "→ donor" = the cache donor's argmax won; "→ kernel" = the arithmetic kernel's own argmax won despite the foreign cache.

Configuration	→ cache donor %	→ kernel %	Other %
Graft A: BF16 model, FP16 cache	47.5	52.5	0.0
Graft B: FP16 model, BF16 cache	35.6	64.4	0.0

BF16 argmax, indicating that the current-step BF16 arithmetic dominates and the cache is only a secondary amplifier.

(2) The cause is asymmetric between precisions. Graft B fixes only 35.6% of flips, about 12pp less than Graft A. The BF16 kernel is more susceptible to the FP16 cache than the FP16 kernel is to the BF16 cache—consistent with BF16's wider rounding error tolerance (storage-vs-computation decomposition, Section 7.6): BF16 arithmetic has a larger "basin" in which the cache can push the argmax; FP16 arithmetic's tighter precision pulls it back to its own argmax even when given a BF16 cache.

(3) Combining graft experiments localises the divergence budget. Putting Tables 8 and 9 together: body-layer hidden-state grafts fix ∼36% of flips (no single layer distinguished), Layer 21 + Final RMSNorm grafts fix 65–71%, KV-cache grafts fix 36–48%, and the BF16 lm_head matmul alone is responsible for the residual ∼21% of "kept BF16" outcomes under RMSNorm graft. These percentages are not additive—graft experiments are not independent treatments—but they delineate the causal surface: *no single upstream component (body layers, KV cache, or final-layer hidden states) is individually sufficient to explain divergence, and the only stage where a minimum-scope FP32 correction can deterministically fix the flip is the lm_head itself*, which is why Intervention C targets exactly there.

(iv) Per-layer top-2 traceback via logit lens

Motivation. The causal picture above localises the flip to the lm_head, but it does not directly answer a natural question: *at the layer before the lm_head, are BF16 and FP16 already disagreeing about which token is most likely?* If so, the lm_head is merely the surface on which an upstream decision becomes visible. If not, the flip is genuinely produced at the lm_head itself. To answer this we run a logit-lens analysis: at each layer ℓ we apply the model's *own* lm_head to the last-token hidden state h_ℓ and record the resulting top-2 tokens, for both BF16 and FP16 paths.

Setup. Same $n = 100$ GSM8K prompts, same measurement step protocol as Section 7.5 (t^* for DIVERGED, matched random step for AGREED). For each layer we report

- `top1_agree`: $\Pr[\arg\max_v \text{lm_head}(h_\ell^{\text{BF16}})_v = \arg\max_v \text{lm_head}(h_\ell^{\text{FP16}})_v]$,
- `top2_jaccard`: Jaccard similarity of the two top-2 sets,
- `bf_top1=final` and `fp_top1=final`: fraction of prompts where the layer-ℓ top-1 already equals the final lm_head top-1 (i.e. the precision has "committed" to its final token by layer ℓ),
- `bf_margin`, `fp_margin`: layer-ℓ top-two gap under each precision.

Three findings sharpen the causal picture. (1) Top-2 token identity is shared along almost the entire forward pass. Under either precision's own lm_head, the top-1 token at *every* body layer ($\ell \leq 20$) agrees between BF16 and FP16 for 93–100% of prompts. The top-2 *set* agrees ≥93% at every layer, including the final lm_head. In other words, upstream of the lm_head the two precisions are not yet disagreeing about "what the likely next tokens are"; they disagree only about how to rank within an otherwise shared pair. This directly answers the diagnostic question: the divergence is not an accumulating token-identity drift; it is a one-shot ranking reversal at the lm_head itself.

Table 10: Per-layer top-2 traceback via logit lens (TinyLlama-1.1B, GSM8K, $n = 100$, selected layers). BF16 and FP16 agree on the top-1 token at every body layer ($\geq 93\%$), and on the top-2 set at every layer ($\geq 93\%$). The disagreement appears only at the final lm_head projection (Layer 21), where DIVERGED-group top-1 agreement drops to 76% while the top-2 set stays 99% aligned. AGREED prompts commit to their final token much earlier (49%/76%/90% at layers 15/18/20) than DIVERGED prompts (7%/34%/39%).

	DIVERGED ($n = 59$)			AGREED ($n = 41$)		
Layer	top1 agree	top2 jac.	bf=final	top1 agree	top2 jac.	bf=final
Embedding	100.0	100.0	0.0	100.0	96.7	0.0
Layer 5	100.0	98.9	0.0	97.6	95.1	7.3
Layer 10	98.3	97.7	1.7	97.6	96.7	17.1
Layer 15	94.9	96.6	6.8	100.0	98.4	48.8
Layer 18	93.2	100.0	33.9	100.0	96.7	75.6
Layer 20	96.6	98.9	39.0	100.0	98.4	90.2
Layer 21 (lm_head)	**76.3**	**98.9**	100.0	**100.0**	**100.0**	100.0
Margins (top-two gap)						
Layer 15 margin	0.075	—	—	0.148	—	—
Layer 18 margin	0.288	—	—	0.921	—	—
Layer 20 margin	0.329	—	—	2.056	—	—
Layer 21 margin	**0.039**	—	—	**5.733**	—	—

(2) Divergence-prone prompts "decide late". The AGREED group commits to its final output token much earlier than the DIVERGED group: `bf=final` reaches 75.6% at layer 18 and 90.2% at layer 20 for AGREED, compared with only 33.9% and 39.0% for DIVERGED. Because DIVERGED prompts are precisely the ones whose ranking has not stabilised by the final body layer, they enter the lm_head with a low margin already baked in. The lm_head projection then either amplifies a stable ranking (AGREED: margin grows from 2.06 to 5.73, $\sim$2.8$\times$ amplification) or fails to stabilise an ambiguous one (DIVERGED: margin shrinks from 0.33 to 0.04, $\sim$8$\times$ *compression*). The lm_head is therefore the stage at which late-decided prompts become precision-sensitive, but the underlying cause is that the upstream trajectory did not arrive at the lm_head with a decisive preference.

(3) This is consistent with the graft experiment (Table 8). Grafting FP16 hidden states into the BF16 forward at any body layer $\ell \leq 20$ changes top-1 only $\sim$36% of the time precisely because BF16 and FP16 body hidden states *already rank the same top token* $\sim$97% of the time (Table 10) and the downstream BF16 kernels on either hidden state preserve that ranking. The +30pp jump at Layer 21 and Final RMSNorm corresponds exactly to the single layer at which the ranking actually changes. The two analyses (graft, logit-lens traceback) triangulate the same conclusion from opposite directions: *the flip is produced at the lm_head; upstream body layers provide the* setup *(an undecided, low-margin trajectory) rather than the* decision.

Cross-task replication. The per-layer top-1 agreement, margin, and commit curves are computed on all three benchmarks (GSM8K, HumanEval, MBPP) with $n = 100$ prompts each. The same pattern appears across these tasks: (i) BF16/FP16 body-layer top-1 agreement stays at 97–98% for both groups and drops only at the lm_head, and only for DIVERGED prompts (to $\sim$90%); (ii) the margin gap opens only at the final layer, with DIVERGED-to-AGREED ratios of 10$\times$ (GSM8K), 14$\times$ (HumanEval), and 9$\times$ (MBPP); and (iii) DIVERGED prompts "decide late" while AGREED prompts commit earlier. This supports the mechanism on the tested math and code tasks; it does not establish task independence beyond them.

Joint interpretation. The three measurements form a causal chain that unfolds across the final transformer layers. The figures quoted in this paragraph are averages over the three benchmarks (GSM8K, HumanEval, MBPP); Table 10 reports the GSM8K-only values, which are sharper at the final layer (margin 0.329 $\rightarrow$ 0.039, top-1 agreement 96.6% $\rightarrow$ 76.3%). At layers 15–19 there is *no differentiation*: both groups show low commit rates ($< 20\%$), moderate margins (0.6–1.4), and near-perfect cross-precision top-1 agreement (97–98%). At layers 19–21 *divergent trajectories emerge*: AGREED prompts begin to commit (margin

rises steeply to 3–4) as their hidden state aligns with a single dominant direction in the lm_head weight space, whereas DIVERGED prompts see their margin *decrease* (from ∼1.4 at layer 19 to ∼1.0 at layer 21) as two candidate tokens compete; cross-precision agreement nonetheless remains $> 95\%$ because the margin, while shrinking, is still $> 40\times$ the per-logit perturbation $\sigma_z \approx 0.026$. At the lm_head the flip occurs: the final RMSNorm plus lm_head projection acts as a *differentiator*, amplifying margins for prompts that already committed ($3 \to 4$) and compressing margins for prompts still undecided ($1.0 \to 0.4$). Once the margin drops to ∼0.4—within ∼15× of σ_z—the BF16/FP16 arithmetic difference becomes comparable to the decision boundary, top-1 agreement falls from ∼97% to ∼90%, and the ∼10% of prompts that flip at this final step cascade into trajectory-level divergence. This three-stage progression—undifferentiated body → emerging competition → lm_head-amplified flip—is consistent across all three tasks and directly implies that *any* intervention targeting layers before the lm_head is premature: the decisive margin compression occurs only at the final projection, which is exactly where Intervention C operates.

III Divergence-conditioned trace on Qwen2.5-3B-Instruct

The lm_head-margin mechanism replicates on Qwen2.5-3B-Instruct (36 layers, d=2048, $|V|$=151,936): top-two margin differs by 90× between groups (0.088 vs. 7.925, 54% vs. 0% flip rate). Body L2 is again indistinguishable.

The two model families support the core mechanism while showing that the body-layer pattern varies by model.

IV Cross-model intervention replication

The headline intervention result (Table 3) is on TinyLlama-1.1B. To test whether the lift transfers across model families and scales, we run the full intervention comparison on four additional models covering three scales and three families: Llama-3.2-3B-Instruct, Qwen2.5-3B-Instruct, DeepSeek-R1-Distill-Qwen-7B, and Mistral-7B-Instruct-v0.3 (GSM8K, $n = 100$, same setup and thresholds as the main experiment). For Mistral-7B we run only Baseline and Intervention C, because Interventions A and B have been shown to be ineffective or redundant with C on the smaller models and doubling the Mistral-7B GPU hours would not change the story; the two-row comparison still gives a matched-pair Pass@1 test (reported in Appendix VI).

The cross-model picture. Table 12 summarises Intervention C's lift across all five tested models, grouped into three tiers by effect size.

Interpretation. (1) Intervention C transfers across scales within the Llama family (Tier 1). The lift is actually *larger* on Llama-3.2-3B (+36pp) than on TinyLlama-1.1B (+22pp), ruling out the hypothesis that the method is a 1B-specific artefact. Both models admit a substantial fraction of prompts in the low-margin, single-step-fixable regime (category B in the per-prompt stratification, Table 4).

(2) Scale alone does not explain the split at 7B (Tiers 2 vs. 3). The two 7B models in our study differ sharply: Mistral-7B-Instruct-v0.3 shows a +8pp EAR lift under Intervention C, while DS-R1-Distill-Qwen-7B shows 0pp. Because both have the same nominal scale, scale alone cannot explain this contrast. The difference is consistent with their observed precision behaviour: the tested Qwen-family models at 1.5B and 7B produce 100% first-token divergence and FP16-body NaNs on GSM8K, placing their divergence before the lm_head and outside Intervention C's reachable scope. Mistral-7B exhibits neither pathology and retains a measurable head-correctable component. This comparison rejects the narrower generalisation that C necessarily fails at 7B, while leaving the underlying model-family cause as a hypothesis.

(3) Tier 2 shares a common profile: Intervention C reconciles a minority of divergences. Three models span Tier 2: Qwen2.5-3B (+3pp), Mistral-7B (+8pp), and OLMoE-1B-7B-MoE (+10pp). The interpretation is consistent across all three and aligns with the mechanism in Section 7.5: a portion of divergence originates in body-layer computation (beyond lm_head scope) and cannot be repaired by any logit-level intervention; only divergences that manifest as late, low-margin lm_head events are reachable by C. The OLMoE data point is particularly informative: the router over MoE experts is a potential

Table 11: Intervention comparison across five additional models (GSM8K, $n = 100$, $\tau = 10^{-3}$). The same ranking holds on every model where the method applies: Integer quantization (A) has zero effect, and Interventions B and C are comparable. Llama-3.2-3B exhibits an even larger absolute lift than the headline TinyLlama numbers (+36pp vs +22pp). Qwen2.5-3B shows a much smaller lift (+3pp), reflecting Qwen's training-time BF16-saturation pattern (Section 6.5). Mistral-7B-Instruct-v0.3, the non-Qwen dense 7B representative, gives a small but clearly non-zero lift (+8pp)—confirming that Intervention C transfers out of the Llama family and to 7B scale when the model is not BF16-saturated, and that the DS-R1-Qwen-7B zero-lift result is a Qwen-family artefact rather than a scale limit. OLMoE-1B-7B, a Mixture-of-Experts architecture (Apache 2.0, Allen AI) with 6.9B total / $\sim$ 1B active parameters per token, gives a +10pp lift, confirming that the method extends beyond dense transformers despite the router-over-experts softmax being a potential extra cross-precision divergence source.

Model	Strategy	EAR (%)	95% CI	Δ vs. Baseline
Llama-3.2-3B-Instruct	Baseline	31.0	[22.8, 40.6]	—
	A. Integer quantization	31.0	[22.8, 40.6]	+0
	B. Top-K FP32	**70.0**	[60.4, 78.1]	+**39**
	C. Full FP32 lm_head	**67.0**	[57.3, 75.4]	+**36**
Qwen2.5-3B-Instruct	Baseline	18.0	[11.7, 26.7]	—
	A. Integer quantization	18.0	[11.7, 26.7]	+0
	B. Top-K FP32	23.0	[15.8, 32.1]	+5
	C. Full FP32 lm_head	21.0	[14.2, 30.0]	+3
Mistral-7B-Instruct-v0.3	Baseline	51.0	[41.4, 60.6]	—
	C. Full FP32 lm_head	**59.0**	[49.2, 68.1]	+**8**
OLMoE-1B-7B-0924-Instruct (MoE)	Baseline	34.0	[25.5, 43.7]	—
	C. Full FP32 lm_head	**44.0**	[34.7, 53.8]	+**10**
DS-R1-Distill-Qwen-7B	Baseline, A, B, C	0.0	[0.0, 3.7]	+0 (Appendix V)

Table 12: Intervention C lift across model families, scales, and architectures (GSM8K, $n = 100$), grouped into three tiers. *Large-lift* models (Tier 1) benefit strongly from C; *small-lift but non-zero* models (Tier 2) still benefit measurably and span three different families (Qwen dense, Mistral dense, OLMoE MoE), showing that Intervention C transfers to the 7B non-Qwen regime (Mistral-7B, +8pp) *and* to sparse MoE architectures (OLMoE, +10pp); the *zero-lift* tier (Tier 3) comprises only Qwen-family BF16-pretrained dense models, which produce FP16 NaNs in body layers and 100% first-token divergence on GSM8K (Section 6.5). *We hypothesise that Intervention C's effectiveness tracks the model's training-time precision stability, rather than parameter count or dense/sparse architecture.* The 7B dense tier is split cleanly by family: Mistral-7B gains +8pp, while DS-R1-Qwen-7B gains 0pp.

Model	Family	Baseline	C EAR	C lift
Tier 1: large lift				
TinyLlama-1.1B-Chat	Llama	41%	63%	+22pp
Llama-3.2-3B-Instruct	Llama	31%	67%	+**36pp**
Tier 2: small but non-zero				
Qwen2.5-3B-Instruct	Qwen	18%	21%	+3pp
Mistral-7B-Instruct-v0.3	Mistral	51%	59%	+8pp
OLMoE-1B-7B-0924-Instruct (MoE)	MoE	34%	44%	+10pp
Tier 3: zero lift (BF16-saturated)				
DS-R1-Distill-Qwen-7B	Qwen	0%	0%	0pp

extra divergence source, since the softmax-over-expert-logits can flip the chosen expert under BF16-vs-FP16 and propagate through different compute paths entirely, yet Intervention C still recovers a +10pp lift—comparable to dense Tier 2 models. This suggests that router flips are either rare on GSM8K at the scale

we test or are themselves concentrated in low-margin steps that C's gate already covers. For all three Tier 2 models, a body-scope method would need to be evaluated to close the residual gap.

(4) The same intervention ranking holds on every model where A, B, C were all run. Integer quantization (A) has exactly zero effect, and Interventions B and C give comparable lifts. The ranking A $<$ B $\approx$ C is a robust empirical finding, independent of model family or scale.

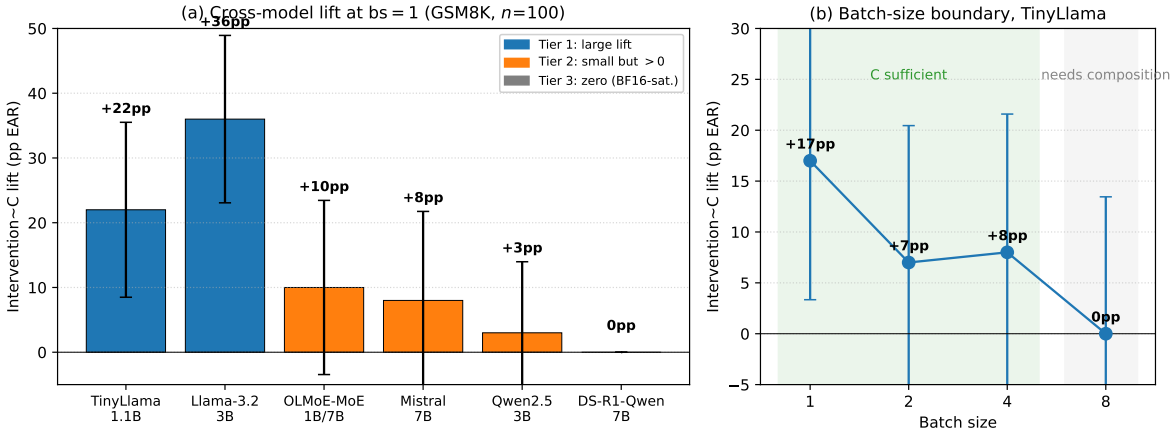

Figure 3: Applicability map for Intervention C. **(a)** Cross-model lift at bs= 1 for six models (including the MoE architecture OLMoE), coloured by tier (Table 12): Tier 1 (Llama family, large lift), Tier 2 (non-Qwen 7B, Qwen-3B, and OLMoE-MoE; small but positive), Tier 3 (Qwen BF16-saturated). Error bars are 95% CIs on the paired lift. **(b)** Batch-size scaling on TinyLlama-1.1B (Appendix VIII): the lift holds at bs$\leq$ 4 (green band, where C alone is sufficient in this experiment) and vanishes at bs$\geq$ 8 (grey band, where composition with global FP32 compute recovers a small positive lift). Together, the panels summarise the tested $(model, bs)$ region in which the lm_head-only intervention helps.

Implications for deployment. Practitioners choosing between Intervention C and body-scope alternatives (e.g., LayerCast) should first measure the target model's divergence rate and margin distribution on the target platform. Our cross-model results suggest three empirical regimes: (i) *Tier 1* (the tested Llama-family models at 1–3B), where C delivers +20 to +36pp EAR; (ii) *Tier 2* (Qwen2.5-3B, Mistral-7B, and OLMoE), where C delivers smaller +3 to +10pp gains; and (iii) *Tier 3* (the tested BF16-saturated Qwen variants), where C is ineffective by itself. These tiers describe the observed models rather than a guaranteed family-level taxonomy. Appendix VIII further reports +17pp at bs= 1, +7–8pp at bs$\in \{2, 4\}$, and 0pp at bs= 8; composition with global FP32 compute recovers +5pp at bs= 8 in the tested setup.

V 7B-scale scope boundary: DeepSeek-R1-Distill-Qwen-7B

Setup. To test whether the gated FP32 lm_head recomputation (Intervention C) transfers to a 7B-parameter model, we rerun the full intervention table (baseline, A, B, C) on DeepSeek-R1-Distill-Qwen-7B, GSM8K, $n = 100$, 256-token budget, $\tau = 10^{-3}$, and the same hardware/software stack as the main experiments. Because the model's BF16 memory footprint is $\sim$14 GB, we run BF16 and FP16 in sequential subprocesses to fit the 22 GB GPU budget; each precision–strategy combination takes $\sim$9.2s per prompt, for a total ablation runtime of $\sim$2 h.

Interpretation: this is a scope boundary, not a method failure. The 0% EAR across all four arms is fully consistent with the paper's analysis and screening data, and we read it as concrete evidence for where Intervention C's scope ends.

(1) The first-divergence step occurs earlier than the intervention's correctable regime. Table 14 breaks down the first-divergence step t^* on the baseline BF16 vs. FP16 outputs. 100/100 prompts diverge,

Table 13: 7B intervention replication (DeepSeek-R1-Distill-Qwen-7B, GSM8K, $n = 100$, $\tau = 10^{-3}$). All four arms collapse to 0% EAR with identical Wilson CIs [0.0, 3.7]%: every prompt diverges, and the interventions cannot recover. Latency columns show per-prompt BF16 / FP16 time; the ∼5% C overhead confirms the intervention is triggering, not silently skipped.

Strategy	Match	EAR (%)	95% CI	BF16 time (s)	FP16 time (s)
Baseline	0/100	0.0	[0.0, 3.7]	9.20	9.22
A (int quant)	0/100	0.0	[0.0, 3.7]	9.40	9.17
B (top-K FP32)	0/100	0.0	[0.0, 3.7]	9.55	9.30
C (full FP32 lm_head)	0/100	0.0	[0.0, 3.7]	9.67	9.21

Table 14: First-divergence step distribution on DeepSeek-R1-Distill-Qwen-7B (GSM8K, $n = 100$ baseline BF16 vs. FP16). Median $t^* = 6$; 86% of prompts diverge in the first 9 tokens, well within the "step-0" regime of Appendix I.

First divergence bucket	Prompts
$t^* = 0$	10
$t^* \in [1, 4]$	21
$t^* \in [5, 9]$	55
$t^* \in [10, 29]$	14
$t^* \geq 30$	0
AGREED (t^* does not exist)	0

with a median $t^* = 6$ and 86/100 prompts diverging within the first 9 tokens. This matches the pattern of Qwen2.5-7B-Instruct on the same benchmark (t^* median = 1, Section 6.5) and sits in the regime Appendix I explicitly identifies as uncorrectable by lm_head-only FP32 recomputation: when divergence happens during the early generation window, the L-layer error accumulated inside the transformer body dominates the single-step lm_head error that Intervention C targets, and the KV cache drifts irrecoverably within a few autoregressive steps.

(2) The intervention *is* firing. Latency for Intervention C on BF16 rises from the baseline's 9.20s to 9.67s (+5%), confirming that FP32 lm_head recomputation is being triggered (the per-step top-two margin does drop below $\tau = 10^{-3}$ on this model, consistent with the small-margin phenomenology in Section 7.5). What fails is not the trigger, but the *correctability* of the resulting flip: by the time a low-margin step arrives, the autoregressive path has already been driven onto a precision-specific KV cache. Appendix II's finding that KV-cache swaps fix only 36–48% of flips *even on TinyLlama* is amplified here, because on this 7B model all 100 prompts sit on the cache-locked branch.

(3) This fits the paper's first-token screening narrative. The model-sensitivity analysis already flags Qwen2.5-7B-Instruct and the Qwen-1.5B family as "catastrophic first-token divergence" models, and our 7B replication extends that list to DeepSeek-R1-Distill-Qwen-7B. These models share an observed inference-time pattern in our experiments: FP16 casting produces unstable, occasionally NaN intermediate activations, and the corresponding argmax trajectories bifurcate within a handful of decoding steps. We hypothesise that training-time precision exposure contributes to this pattern, but we do not verify the models' training histories or establish that causal mechanism. For this class of models, an lm_head-scoped intervention is not sufficient; the present experiment establishes only that Interventions A–C cannot recover agreement. Our TinyLlama-1.1B and Qwen2.5-3B-Instruct results (Section 7.5, Appendix III), which do admit a non-empty AGREED group and do respond to Intervention C, characterise where the method applies; the 7B result here characterises where it does not.

Practical takeaway. Practitioners deploying a 7B-scale BF16-saturated model should not expect Intervention C (or any other inference-time lm_head-scoped fix) to reconcile BF16-vs-FP16 outputs. The first reproducibility measure for such models is to pin the serving precision. If cross-precision reproducibility remains required, a body-scope approach such as LayerCast or precision-aware retraining would need to be

Table 15: Per-precision task accuracy on GSM8K ($n = 100$, final-answer extraction). McNemar exact paired tests compare Baseline vs. Intervention C within each (model, precision) cell. No test rejects the null at any conventional level. Despite the Llama-3.2-3B family showing the largest EAR lift in the paper (+36pp, Table 11), its task accuracy is unchanged under Intervention C; despite Mistral-7B being the only 7B non-Qwen model we tested, its Pass@1 point estimates drop by 3–5pp but with non-significant McNemar p values (0.125 and 0.250) and substantially overlapping Wilson CIs. The six within-model tests (two per model) span EAR lifts from +3pp to +36pp and provide no evidence that Intervention C systematically shifts either trajectory toward a wrong answer.

Model	Config	Correct	Acc (%)	95% CI	vs. Baseline
Qwen2.5-3B	BF16, Baseline	27/100	27.0	[19.3, 36.4]	—
	BF16, Intervention C	29/100	29.0	[21.0, 38.5]	ΔAcc= +2pp ($p = 0.688$)
	FP16, Baseline	28/100	28.0	[20.1, 37.5]	—
	FP16, Intervention C	27/100	27.0	[19.3, 36.4]	ΔAcc= −1pp ($p = 1.000$)
Llama-3.2-3B	BF16, Baseline	24/100	24.0	[16.7, 33.2]	—
	BF16, Intervention C	23/100	23.0	[15.8, 32.1]	ΔAcc= −1pp ($p = 1.000$)
	FP16, Baseline	20/100	20.0	[13.3, 28.9]	—
	FP16, Intervention C	21/100	21.0	[14.2, 30.0]	ΔAcc= +1pp ($p = 1.000$)
Mistral-7B	BF16, Baseline	43/100	43.0	[33.7, 52.8]	—
	BF16, Intervention C	38/100	38.0	[29.1, 47.8]	ΔAcc= −5pp ($p = 0.125$, +1/ − 6)
	FP16, Baseline	40/100	40.0	[30.9, 49.8]	—
	FP16, Intervention C	37/100	37.0	[28.2, 46.8]	ΔAcc= −3pp ($p = 0.250$, +0/ − 3)

evaluated; we do not test those alternatives on this model. Our main-text method is designed for—and evaluated on—the substantial class of models where the divergence is a late, small-margin event at the lm_head, which is exactly what makes it a single-step-correctable event.

VI Quality preservation: does Intervention C change task accuracy?

Setup. Intervention C reconciles BF16-vs-FP16 outputs at the lm_head, but a natural concern is whether this comes at the cost of *task accuracy*: perhaps the intervention pushes both precisions toward the same *wrong* answer. We test this directly on GSM8K ($n = 100$) by running four configurations—BF16 baseline, FP16 baseline, BF16 + C, FP16 + C—and extracting the final numerical answer from each generation. We use three testbeds spanning the three family/scale tiers of Table 12: Qwen2.5-3B-Instruct (Tier 2, Qwen-family), Llama-3.2-3B-Instruct (Tier 1, largest EAR lift in the paper), and Mistral-7B-Instruct-v0.3 (Tier 2, non-Qwen 7B). All three models have GSM8K zero-shot accuracy well above the $\sim$3% noise floor we observe on TinyLlama,[3] giving matched-pair tests enough statistical power to detect small accuracy shifts.

Findings. (1) No statistically significant accuracy change in either direction, on any model. Under matched-pair McNemar tests, no (model, precision) cell rejects the null at any conventional level. For Qwen2.5-3B, BF16 goes +4/ − 2 ($p = 0.688$) and FP16 +1/ − 2 ($p = 1.000$). For Llama-3.2-3B, BF16 goes +8/ − 9 ($p = 1.000$) and FP16 +2/ − 1 ($p = 1.000$). For Mistral-7B, BF16 goes +1/ − 6 ($p = 0.125$) and FP16 +0/ − 3 ($p = 0.250$); both p-values sit above the 0.05 threshold but below 0.3, and the point-estimate drops (−5pp on BF16, −3pp on FP16) are worth a dedicated discussion (see finding (3) below). The paper's +22pp (TinyLlama), +36pp (Llama-3.2-3B), +3pp (Qwen2.5-3B), and +8pp (Mistral-7B) EAR gains are therefore not artefacts of Intervention C systematically pushing both precisions toward the same wrong answer.

[3] All reported accuracies in this section use our bare-prompt format (`"Solve step by step:\n{question}\nAnswer:"`) with greedy decoding and no system prompt or chat template. This choice is deliberate: it is the same format used for the cross-family EAR comparison (Table 11) and keeps the setup invariant across TinyLlama, Qwen, Llama, and Mistral families, so the EAR numbers are directly comparable. Absolute Pass@1 under this format is a lower bound on what each model can achieve with its native chat template. Because the quality-preservation tests are matched-pair McNemar tests over the same prompts in all four arms, the conclusions about Intervention C's accuracy effect are unaffected by prompt-format choice; a chat-template replication confirms this empirically by repeating the Llama-3.2-3B rows with the native chat template, where absolute Pass@1 rises to 66% but the McNemar tests remain non-significant ($p = 0.625$ and $p = 1.000$).

Table 16: Prompt-format robustness: the same four-arm experiment on Llama-3.2-3B-Instruct using its native chat template with a short system prompt. Absolute Pass@1 more than doubles (24%→66% on BF16 baseline), confirming that the lower numbers in the quality-preservation results reflect prompt format rather than model capability. Both McNemar paired tests remain non-significant, and FP16 shows zero Pass@1 flips under Intervention C ($+0/-0$): the intervention reconciles outputs exactly. The quality-preservation conclusion is therefore invariant to prompt format.

Config	Correct	Acc (%)	95% CI	vs. Baseline
BF16, Baseline	66/100	66.0	[56.3, 74.5]	—
BF16, Intervention C	64/100	64.0	[54.2, 72.7]	ΔAcc= −2pp (p=0.625, $+1/-3$)
FP16, Baseline	66/100	66.0	[56.3, 74.5]	—
FP16, Intervention C	66/100	66.0	[56.3, 74.5]	ΔAcc= 0pp (p=1.000, $+0/-0$)

(2) BF16 and FP16 baselines have essentially the same task accuracy within each model. On Qwen2.5-3B, BF16 baseline Pass@1 is 27% and FP16 baseline is 28%; on Llama-3.2-3B, BF16 is 24% and FP16 is 20%; on Mistral-7B, BF16 is 43% and FP16 is 40%. The within-model gaps ($\leq$ 4pp) are within paired-sample uncertainty. Aggregate task accuracy therefore does not identify a uniformly preferable low-precision arm on disagreeing prompts. The separate FP32-fidelity analysis (Appendix XIII) shows that FP16 is closer to the FP32 trajectory overall; Intervention C is intended to improve cross-format replay when the serving format cannot simply be changed.

(3) Effect-size patterns across tiers: large EAR lift with balanced churn (Llama-3.2-3B) and small EAR lift with an uncertain accuracy shift (Mistral-7B). Llama-3.2-3B (Tier 1, +36pp EAR) exhibits large but balanced within-precision correctness churn under C ($+8/-9$ on BF16). Mistral-7B (Tier 2, +8pp EAR) exhibits asymmetric small churn ($+1/-6$ on BF16, $+0/-3$ on FP16) that does not reach significance but points in one direction. The data are compatible with either sampling noise or a small negative accuracy effect; at $n=100$ we cannot distinguish them. Consequently, a deployment should evaluate both agreement and task quality on its own workload rather than assuming that a modest agreement gain justifies an uncertain quality trade-off. A larger-n Mistral-7B follow-up remains useful.

(4) Small-model baseline for comparison. We also ran the same experiment on TinyLlama-1.1B (Pass@1 around 3–4% for all four arms, $n = 100$). Matched-pair McNemar tests are again non-significant ($p > 0.3$). However TinyLlama's accuracy sits at the noise floor for $n=100$ and the tests have limited power; the Qwen2.5-3B, Llama-3.2-3B, and Mistral-7B rows of the quality-preservation results are the primary quality-preservation evidence.

(5) Implication for the main claim. The quality results show that Intervention C improves BF16-vs-FP16 output agreement (+22pp EAR on TinyLlama, Table 3; +36pp on Llama-3.2-3B, Table 11; with scaled agreement replication in Appendix X) without a statistically significant accuracy change on the four tested models. This is evidence of quality preservation in the tested conditions, not proof of zero effect: the Mistral point estimates and the finite sample sizes motivate workload-specific validation.

Prompt-format robustness check (Llama-3.2-3B). The Llama-3.2-3B Pass@1 numbers in the quality-preservation results (20–24%) sit well below the model's advertised GSM8K accuracy because the bare-prompt format used throughout our cross-family experiments does not invoke the chat template against which Llama-3.2-3B-Instruct was tuned. To confirm that the quality-preservation conclusion does not depend on this choice, we repeat the four-arm experiment with Llama-3.2-3B's native chat template (using a short system prompt: "You are a careful math tutor. Solve the problem step by step. End your response with a line that contains the final numeric answer only."), keeping every other setting identical (same 100 GSM8K test items, greedy decoding, $\tau = 10^{-3}$, max_new_tokens = 512).

The chat-template results show two patterns. First, the FP16 arm produces *identical* per-prompt correctness under Baseline and Intervention C ($+0/-0$): on the 66 prompts where FP16 answered correctly under baseline it still does so under C, and symmetrically for the incorrect prompts. The intervention therefore preserves which FP16 prompts are answered correctly while reconciling their token trajectories with the

Table 17: Long-reasoning sanity check on MATH-500 ($n = 50$, max_new_tokens = 1024, bare-prompt format). Mean generated length is 430–610 tokens per prompt, 3–4× longer than the main-text GSM8K chains. Baseline EAR remains close to the GSM8K values, and Intervention C has a positive point-estimate lift on both models. The per-cell Wilson intervals are descriptive rather than paired intervals for the lifts.

Model	Variant	EAR (%)	95% CI	Avg. len	Trig/step
TinyLlama-1.1B-Chat	Baseline	44.0	[31.2, 57.7]	606	0.00
	Intervention C	64.0	[50.1, 75.9]	564	0.82
Llama-3.2-3B-Instruct	Baseline	30.0	[19.1, 43.8]	438	0.00
	Intervention C	56.0	[42.3, 68.8]	430	1.89

BF16 counterpart. Second, the BF16 arm shows a 2pp drop ($-3/+1$, McNemar $p=0.625$), well within noise and with substantially overlapping Wilson CIs. Both arms are individually and jointly consistent with no accuracy change. Combined with the main quality-preservation table, we therefore have quality-preservation evidence on (Qwen2.5-3B, Llama-3.2-3B, and Mistral-7B with bare prompt, plus Llama-3.2-3B with chat template), across four model / prompt-format conditions, none of which detect a significant accuracy effect from Intervention C.

VII Long-reasoning sanity check: does divergence grow with chain length?

Motivation. The main-text GSM8K experiments cap generation at 256 tokens, which matches the typical reasoning-chain length for GSM8K problems. Modern LLM workloads, however, routinely generate chains of 500–4000 tokens (competition math, agentic planning, long-context tool use). A natural question is whether the 30–64% cross-precision divergence we report is a short-chain artefact that would either amplify or vanish under longer generations. The a priori prediction was amplification: longer chains contain more low-margin steps, each one an opportunity for BF16 and FP16 to fork, and a single fork cascades. This appendix tests that prediction directly on MATH-500, a harder benchmark (Hendrycks-style competition math) at max_new_tokens = 1024.

Setup. We draw the first $n = 50$ MATH-500 test problems deterministically. Prompt format is bare (`"Problem:\n{problem}\n\nSolution:\n"`), consistent with every other cross-family experiment in the paper—no chat template, no system prompt. We run baseline and Intervention C on TinyLlama-1.1B-Chat and Llama-3.2-3B-Instruct (the two Tier 1 models from Table 12) in both BF16 and FP16, batch size 1, fixed seed, A10G 22 GB.

Findings. (1) Baseline divergence does not amplify with chain length. Compared to the GSM8K numbers in Table 11 (TinyLlama baseline 41%, Llama-3.2-3B baseline 31%), the MATH-500 baselines are 44% and 30% respectively—both within 3pp of their GSM8K counterparts despite the mean chain length tripling (170 tokens on GSM8K → 438–606 tokens on MATH-500) and the benchmark difficulty increasing substantially. This rules out the "long chain cascades more" hypothesis at the lengths we test.

(2) The saturation is consistent with the margin-based mechanism. Section 7.5 shows that divergence is associated with the density of low-margin lm_head steps rather than chain length alone. MATH-500 has more gated steps per prompt than GSM8K (trigger rate 1.89 vs. ∼1 on Llama-3.2-3B GSM8K), yet prompt-level divergence remains similar. Once a token differs, the autoregressive trajectories usually remain different; before that first differing token, additional decoding steps create further opportunities for a low-margin fork. Over the tested length range, the observed rates suggest that this prompt-level probability has already approached a plateau.

(3) Intervention C's point-estimate lift persists at longer chains. The lifts are +20pp (TinyLlama) and +26pp (Llama-3.2-3B) on MATH-500, compared with +22pp (TinyLlama, Table 3) and +36pp (Llama-3.2-3B, Table 11) on GSM8K. The MATH-500 point estimates are positive and of similar order, but the $n=50$ per-cell Wilson intervals are descriptive and do not constitute paired confidence intervals for the lifts.

The Llama-3.2-3B point estimate is lower on MATH-500; determining whether this reflects task difficulty, trigger density, or sampling variation would require a larger paired evaluation.

(4) An incidental latency observation. On MATH-500, Llama-3.2-3B's FP16 baseline wall-clock time (10.93s/prompt) is about 24% *faster* than its BF16 counterpart (14.46s/prompt). This is the first latency observation of this magnitude in our experiments and is not relevant to the intervention claim, but we record it here for transparency. The most plausible explanation is that SM 86 (A10G) has a somewhat faster FP16 FMA path than its BF16 FMA path, plausibly amplified by the longer generation lengths on MATH-500; we did not pursue this further.

Summary. The phenomenon and a positive Intervention C point-estimate lift both persist on chains that are 3–4× longer. The Wilson intervals are wider at $n=50$ than in the $n=100$ GSM8K comparison, so these results support qualitative transfer rather than equality of effect sizes. Over the range explored, divergence does not increase with reasoning length alone.

VIII Production-stack robustness

Table 18: Production-stack robustness checks (TinyLlama-1.1B, GSM8K).

Axis	Result	Implication
MoE (OLMoE)	100% BF16 self-consistency	Kernel non-det. ruled out
FlashAttention-2	EAR −1pp; C lift +18pp (= eager)	Consistent across both tested kernels
Batch size	C lift: +17/+7/+7/0pp (bs 1/2/4/8)	Bounded to bs≤4
Global FP32 compute + C	+5pp at bs=8; noise at bs=16	Composition viable
Overhead	+1.4% latency, +11% memory	Negligible cost

IX Hyperparameter and intervention ablations

Table 19: Ablation summary.

Ablation	Result	Confirms
Scope: RMSNorm+lm_head	EAR drops 63%→44%	lm_head-only optimal
Threshold $\tau \in [10^{-4}, 10^{-2}]$	EAR insensitive	Bimodal margin
Top-K: $K \in \{2, ..., \lvert V \rvert\}$	All identical EAR	Rank-1-vs-2 only
Temperature sharpening	Zero effect	Scale-invariant ratio
Consensus decoding	100% at 2× cost	Trivial upper bound

X Statistical validation and reproducibility

Wilson 95% CIs have half-width ±9–10pp at n=100 and describe the individual agreement rates. Paired bootstrap intervals describe the lifts; for example, the TinyLlama +22pp lift has a 90% CI of [+11, +33]pp. Scaled replication at n=300 (GSM8K), n=164 (full HumanEval), and n=257 (full MBPP) confirms all n=100 point estimates fall inside the scaled CIs; the scaled lifts are +16.4pp (GSM8K), +28.1pp (HumanEval), and +27.2pp (MBPP), each with McNemar $p < 0.001$, with per-cell CI half-widths reduced to ±5.5–7.4pp.

XI Reproducibility statement

Environment and methodology. The primary experiments use an NVIDIA A10G (22 GB, Ampere SM 86) under PyTorch 1.13 / 2.1 with CUDA 11.7. Cross-hardware replications use an NVIDIA L4 (24 GB, Ada SM 89) and A100-SXM4 (40 GB, Ampere SM 80); the FP16/FP32 probe uses a T4 (16 GB, Turing SM 75). We enable deterministic flags (`torch.use_deterministic_algorithms(True)`, fixed CUBLAS

workspace) and use greedy decoding (T=0) with fixed prompt selection. Models are loaded from Hugging Face Hub without checkpoint modification, except for the explicitly described runtime precision transformations. EAR compares complete token-id sequences; task-level agreement uses final-answer extraction for GSM8K and execution outcomes for HumanEval. Confidence intervals are Wilson score intervals unless otherwise stated, and bootstrap CIs use 10,000 resamples. The prompt formats, sample sizes, hardware exceptions, and intervention hyperparameters (τ=10^{-3} for BF16/FP16 and K=8 for Intervention B) are reported in the corresponding sections; the FP8 threshold is separately calibrated as described in Appendix XV.

XII Directional body-error analysis

To test whether body-layer error contributes directionally to flips, we project the BF16-vs-FP16 logit-error vector onto the top-two decision direction $d = e_{v^{(1)}} - e_{v^{(2)}}$. For the empirical top-two decision rule evaluated here, we compare the magnitude of this projection with the smaller top-two margin across the two arms; exceeding that margin predicts a flip. This rule approximates the exact all-competitor condition in Proposition 1. Note on populations: this analysis measures margins at the *first-divergence step* t^* of each prompt (median 0.0 for DIVERGED), whereas Table 7 reports margins at each group's fixed measurement step under the divergence-conditioned trace protocol (150× gap); the two tables therefore quote different statistics of different step populations, and both show the same qualitative separation of two-plus orders of magnitude.

Table 20: Directional body-error analysis (GSM8K, n=100 per model). The mean body projection onto the top-two direction is indistinguishable between Diverged and Agreed steps on both models, confirming that body error does not preferentially point toward the decision boundary for flipping steps. The margin, not the directional body projection, separates the two groups.

Model	Quantity	DIVERGED	AGREED	Interpretation
TinyLlama-1.1B	Hidden-state L2 $\|e\|$	0.999	1.052	Indistinguishable
	Body projection (mean ± SD)	0.025 ± 0.023	0.026 ± 0.020	Equiv. (p=0.014)
	Signed dir. perturb. $\Delta z^{(1)} - \Delta z^{(2)}$	0.039	0.036	Indistinguishable
	Top-two margin (mean)	0.015	3.93	260× gap
	Top-two margin (median)	0.0	2.69	—
Qwen2.5-3B	Body projection (mean ± SD)	0.051 ± 0.070	0.049 ± 0.043	Equiv. (p=0.006)
	Top-two margin (mean)	0.035	6.05	170× gap

The body-error null is formally established, not merely unrejected. Because a null result carries weight only if it is powered, we test equivalence rather than relying on a failure to reject. On TinyLlama-1.1B the difference in body projection between DIVERGED and AGREED is −0.0013 (95% CI [−0.0099, +0.0073]; Welch $t = -0.30$, $p = 0.77$), and the two one-sided tests (TOST) procedure, with an equivalence margin of half a pooled SD (0.011 for TinyLlama, 0.032 for Qwen), rejects non-equivalence at $p = 0.014$. On Qwen2.5-3B the difference is +0.0015 (95% CI [−0.0216, +0.0247]; $t = +0.13$, $p = 0.90$), with TOST $p = 0.006$. We use a half-SD margin as a conventional medium effect size; the point estimates differ by < 0.06 SD, far inside it. So the two groups are statistically *equivalent* on the decision-direction projection, while their top-two margins differ by two to three orders of magnitude (0.015 vs. 3.93; 0.035 vs. 6.05). The asymmetry between these two comparisons is the substantive content of this appendix.

A note on what these accuracies measure. Proposition 1 is exact, so it cannot have an error rate; the quantity it involves—the per-coordinate perturbation at the step in question—is not observable before the step is computed. What we evaluate here is therefore a *decision rule* derived from it: predict a flip when $|\Delta z(v^{(1)}) - \Delta z(v^{(2)})|$ exceeds the smaller of the two arms' margins, thresholding the *magnitude* $|\Delta z(v^{(1)}) - \Delta z(v^{(2)})|$ against the margin. Under this rule the directional statistic classifies flip-vs-no-flip correctly on 88% of steps for TinyLlama-1.1B and 89% for Qwen2.5-3B, versus step-level base rates of 63%/55% and versus 86%/83% for the RMS statistic it replaces. (The magnitude, not the signed difference, is the right predictor here: the signed rule scores only 0.57/0.59, because a flip can be triggered by a perturbation of either sign relative to the current top-1.) The exact condition of Proposition 1 is proved; this decision rule

is the deployable approximation to it, and the residual error is why we label the scale analysis heuristic. The intervention's own gate does not use this statistic at all—it fires on the observable margin $\Delta_t < \tau$; the decision rule here is only an analysis tool for attributing flips. This confirms that (i) the lm_head margin is the dominant factor, (ii) body error does not preferentially align with the decision boundary, and (iii) the directional formulation of Proposition 1 is empirically supported.

XIII FP32 fidelity of Intervention C

To verify that Intervention C pushes outputs toward the FP32-oracle trajectory (rather than toward an arbitrary third path), we measure agreement between Intervention C outputs and FP32 greedy outputs, comparing against the BF16-baseline-to-FP32 agreement rate.

Table 21: FP32 reference fidelity (GSM8K, n=100). Three metrics: (1) sequence-level agreement with the FP32 oracle, (2) repaired-token agreement with the FP32 argmax at triggered steps, and (3) downstream-correctness alignment: the fraction of FP32-correct prompts that are also answered correctly under each method. Intervention C raises sequence agreement with FP32 by +19–21pp. Brackets on the sequence-agreement rows are 95% CIs for the *lift*; on the repaired-token rows they are Wilson 95% CIs for the proportion.

Model	Metric	BF16→FP32	FP16→FP32	IntC→FP32
TinyLlama-1.1B	Sequence agreement	42%	90%	**63**% (+21pp) [+7, +35]
	Repaired-token = FP32 argmax	—	—	63% (97/154) [55, 70]
	Correctness alignment w/ FP32	50%	100%	50%
Qwen2.5-3B	Sequence agreement	23%	82%	**42**% (+19pp) [+6, +32]
	Repaired-token = FP32 argmax	—	—	51% (132/260) [45, 57]
	Correctness alignment w/ FP32	25%	92%	**50**%

On both models, Intervention C substantially increases FP32 fidelity: sequence agreement rises from 42% to 63% on TinyLlama (+21pp) and from 23% to 42% on Qwen2.5-3B (+19pp). At triggered steps, 63% (TinyLlama) and 51% (Qwen) of repaired tokens match the FP32 argmax. Downstream-correctness alignment with FP32 doubles on Qwen2.5-3B (25% → 50%); on TinyLlama the alignment is unchanged at 50%, though we caution that only 2 of 100 prompts are FP32-correct on this model (GSM8K accuracy ∼2%), so this cell carries little statistical weight. FP16 already shows high FP32 fidelity (90%/82%) because FP16's 10-bit mantissa introduces smaller rounding errors; Intervention C moves BF16 outputs measurably toward this higher-precision reference rather than toward an arbitrary consensus. Two qualifications are important. First, FP16 alone attains higher FP32 fidelity than BF16+C: if maximal FP32 fidelity is the objective and the serving format is free, the appropriate action is to serve FP16. Our claim is narrower—within a fixed BF16 deployment, the intervention moves outputs toward the FP32 reference. Second, because a repaired step is a choice between exactly two candidates, the chance level for repaired-token agreement is 50%: TinyLlama's 63% (CI [55, 70]) is above chance, whereas Qwen's 51% (CI [45, 57]) is not distinguishable from it. Per-token fidelity is therefore established on TinyLlama only; on Qwen the sequence-level (+19pp) and correctness-alignment (25% → 50%, on 12 FP32-correct prompts) gains are the operative evidence.

XIV Cross-architecture replication: L4 (Ada), A100 (Ampere), and T4 (Turing)

To verify that the mechanism and intervention are not A10G-specific, we replicate the headline experiment on an NVIDIA L4 (Ada Lovelace, SM 89, 24 GB) and an NVIDIA A100-SXM4-40GB (Ampere, SM 80), and additionally probe the Turing generation via an NVIDIA T4 (SM 75, 16 GB).

The intervention transfers to every microarchitecture we tested, but the magnitude of its benefit does not, and we state that plainly. Two readings follow from the three-GPU comparison.

First, *the direction of the effect is robust; the size is not.* All six model×GPU cells show a positive lift, but TinyLlama ranges from +22pp (A10G) to +12pp (A100) and Qwen2.5-3B from +3pp (A10G) to +20pp

Table 22: Cross-architecture replication (GSM8K, n=100, τ=10^{-3}). The intervention yields a positive lift on all three GPU microarchitectures tested; the *magnitude* is hardware-dependent, and A10G is the most favourable card for TinyLlama and the least favourable for Qwen2.5-3B.

Model	GPU (architecture)	Baseline EAR (%)	IntC EAR (%)	Lift
TinyLlama-1.1B	A10G (Ampere, SM 86)	41	63	+22pp
	L4 (Ada, SM 89)	36	57	+21pp
	A100 (Ampere, SM 80)	39	51	+12pp
Qwen2.5-3B-Instruct	A10G (Ampere, SM 86)	18	21	+3pp
	L4 (Ada, SM 89)	30	50	+20pp
	A100 (Ampere, SM 80)	20	34	+14pp

(L4). The headline +22–36pp figures reported elsewhere in this paper are therefore A10G measurements and should be read as one point in a hardware-dependent range, not as architecture-independent constants.

Second, *the A10G result is the outlier in this three-GPU comparison.* Qwen's lift is +3pp on A10G, +14pp on A100, and +20pp on L4. Because A100 (SM 80) and A10G (SM 86) are both Ampere yet differ by 11pp, and because the two non-A10G cards agree more closely with each other, the variation cannot be attributed simply to the Ada architecture or the Qwen checkpoint. One possible mechanism is that A10G's FP16 kernel path pushes more activations outside the low-margin, lm_head-reachable regime; we have not instrumented the kernels to test this explanation.

The practical consequence is that the tier assignments and the CV(TD) screening metric elsewhere in this paper are calibrated on A10G and require per-platform recalibration; Qwen2.5-3B is Tier 2 on A10G but would be Tier 1 on L4.

A third generation: T4 (Turing). The T4 cannot run this paper's primary comparison at all: Turing (SM 75) predates native BF16 support, which begins with Ampere (SM 80). Running BF16 there would measure software emulation rather than hardware format behaviour, and FP8 requires SM 89. The BF16/FP16 comparison therefore applies to Ampere-and-later serving stacks; on T4, FP16-vs-FP32 is the meaningful cross-precision pair, which we ran (TinyLlama-1.1B, GSM8K, n=100, τ=10^{-3}):

Table 23: T4 (Turing, SM 75) FP16-vs-FP32 replication. Because FP32 holds the *original* weights, this pair carries no weight-truncation floor, and baseline agreement is correspondingly far higher than in the BF16-vs-FP16 setting (88% vs. 41%). Gated FP32 lm_head recomputation still yields a positive lift while triggering on only 20 steps.

Precision pair	Baseline EAR (%)	IntC EAR (%)	Lift
FP16 vs. FP32 (T4)	88	95	+7pp

Two observations. First, the much higher baseline agreement is what the storage-vs-computation decomposition predicts: FP32 is the untruncated weight, so the ∼12pp irreducible floor identified for BF16-vs-FP16 (Section 7.6) does not apply here, and the residual disagreement is arithmetic-only. Second, the intervention remains effective in this regime (+7pp from 20 triggered steps), so the low-margin mechanism is not an artefact of any single format pair or microarchitecture: across Turing, Ampere, and Ada, and across three precision pairs (FP16/FP32, BF16/FP16, BF16/FP8), gated FP32 lm_head recomputation improves exact agreement. We report a single model here, as the T4's 16 GB limits FP32-arm capacity.

XV FP8 replication and threshold scaling

To test whether the mechanism and intervention extend beyond the BF16/FP16 pair to a lower-precision serving format, we replicate the headline experiment with FP8. We run on an NVIDIA L4 GPU (Ada Lovelace, compute capability 8.9, natively FP8-capable) and apply FP8-E4M3 with per-tensor dynamic

scaling to the lm_head projection, keeping the transformer body in BF16 in *both* arms so that the comparison isolates the head-projection format—consistent with the finding that the lm_head is the decisive locus (Section 7). The setup otherwise matches the main experiments: GSM8K, n=100, greedy decoding, 256-token budget, on TinyLlama-1.1B-Chat and Qwen2.5-3B-Instruct.

Table 24: Baseline FP8 divergence (GSM8K, n=100; lm_head in FP8-E4M3 vs. BF16, transformer body in BF16 in both arms). FP8 divergence is substantially more severe than the FP16 setting (70% vs. 59% of prompts on TinyLlama; 93% vs. 75% on Qwen), and flips still concentrate at low-margin steps. The FP16 reference column uses the same bare-prompt protocol as this experiment (see the cross-task appendix footnote for the protocol difference from the main-text chat-template numbers).

Model	EAR (BF16 vs FP8)	EAR (BF16 vs FP16) [reference]	First-div margin (median)
TinyLlama-1.1B	30%	41%	0.0625
Qwen2.5-3B	7%	25%	0.125

FP8's 3-bit mantissa produces larger rounding perturbations than FP16's 10 bits, so divergence rises on both models; yet the first-divergence margins remain small relative to the format's perturbation scale, showing that the low-margin lm_head pattern also appears in the tested head-isolated BF16/FP8 setting.

Threshold scaling. The FP16-calibrated trigger threshold $\tau = 10^{-3}$ *fails* under FP8: the gate opens on only 3 of ∼25,600 decoding steps, and the lift (−1pp) is within noise. This failure is predicted by the scale analysis accompanying Proposition 1: FP8-E4M3's per-logit perturbation scale is roughly 2–5× the FP16 scenario's $\sigma_z \approx 0.026$, so a threshold calibrated to the FP16 perturbation scale sits far below the margins at which FP8 flips occur. The trigger threshold must scale with the format's perturbation scale. Rescaling τ recovers and then completes the repair (TinyLlama, both-arms gated protocol identical to the main experiments):

Table 25: Threshold sweep for gated FP32 lm_head recomputation under FP8 (GSM8K, n=100 per model, both-arms gated protocol). The FP16-calibrated $\tau = 10^{-3}$ triggers almost never on either model; rescaling τ to the FP8 perturbation scale monotonically recovers agreement on both, reaching exact agreement on TinyLlama (100%) and 63% on Qwen at $\tau = 0.25$, each at a ∼4% trigger rate. Bracketed values are Wilson 95% CIs; the $\tau = 0.25$ lifts are +70pp (CI [+61, +79]) and +56pp (CI [+45, +67]).

	TinyLlama-1.1B			Qwen2.5-3B		
τ	EAR (%)	Lift (pp)	Trig. steps	EAR (%)	Lift (pp)	Trig. steps
10^{-3}	29 [21, 39]	−1	3	10 [6, 17]	+3	4
0.02	37 [28, 47]	+7	73	13 [8, 21]	+6	94
0.05	48 [38, 58]	+18	201	16 [10, 24]	+9	197
0.1	77 [68, 84]	+47	378	27 [19, 36]	+20	396
0.25	**100** [96, 100]	+70	1010 (∼4%)	**63** [53, 72]	+56	994 (∼4%)

Why the threshold must scale: the flip-margin distribution moves with the format. The scale analysis predicts that flips concentrate where the margin is small *relative to the format's perturbation scale*—so a more aggressive format should produce flips at correspondingly *larger* margins. This is directly observable in the first-divergence margins:

The shift is quantitative, not merely directional: relative to the FP16-setting scale $\sigma_z \approx 0.026$, the FP8 flip-margin medians sit at 2.4× (TinyLlama) and 4.8× (Qwen) that value, consistent with the 2–5× larger per-logit perturbation of FP8-E4M3. The gate must widen by a comparable factor, which is what the sweep finds empirically (τ from 10^{-3} to 0.25). We note this is a two-format comparison; whether the relationship is proportional across a wider range of formats is untested.

At $\tau = 0.25$ the intervention achieves *exact* agreement (EAR 100%) on TinyLlama at a ∼4% trigger rate—comparable overhead to the FP16 setting. The rescaling behaviour replicates on Qwen2.5-3B: the same sweep

Table 26: Top-two margin at the first divergence step, by format pair (GSM8K, n=100 per cell). Under FP8 the flips occur at margins 2.4–4.8× larger than under FP16, matching the larger per-logit perturbation of the 3-bit-mantissa format and explaining why a threshold calibrated for FP16 ($\tau = 10^{-3}$) almost never fires under FP8.

Format pair	Model	Flip-margin median	Flip-margin mean
BF16 vs. FP16	TinyLlama-1.1B	0.000	0.015
BF16 vs. FP16	Qwen2.5-3B	0.000	0.035
BF16 vs. FP8	TinyLlama-1.1B	**0.0625**	0.052
BF16 vs. FP8	Qwen2.5-3B	**0.1250**	0.126

recovers monotonically from +3pp at $\tau = 10^{-3}$ to +56pp at $\tau = 0.25$ (EAR 7% → 63%, at a comparable ∼4% trigger rate), though it does not reach exact agreement. Note that the explanation cannot be body-originated divergence, which this design excludes: with identical BF16 bodies in both arms the shortfall must come from gate coverage. Qwen's FP8 flip-margin distribution has a heavier tail (median 0.125 versus TinyLlama's 0.0625, Table 26), so $\tau = 0.25$ leaves more of its flippable steps ungated; a larger τ would be needed, at proportionally higher trigger cost.

End-to-end FP8: where the method's reach ends. The configuration above deliberately isolates the head projection. We also ran the deployment-shaped variant, quantising *every* `nn.Linear` in the model to FP8-E4M3 (155 modules; TinyLlama-1.1B, GSM8K, n=100). Divergence becomes far more severe — EAR 10%, i.e. 90% of prompts diverge — and the gated repair recovers only +1pp. Widening the gate does not help: $\tau = 0.1$ and $\tau = 0.25$ trigger on 321 and 738 steps respectively and both yield the same +1pp. This is the boundary the error model predicts rather than a contradiction of it: with all linear layers quantized, divergence is dominated by body-originated error accumulated across the forward pass, which lies outside the reach of any lm_head-scope repair. It is the same limit we report for BF16-saturated Qwen variants (Appendix V) and for bs≥8 batched decode (Appendix VIII): where lm_head-originated divergence is not the dominant source, composition with a body-scope method is required. The contrast between the two FP8 configurations (+70pp when the body is held fixed, +1pp when it is not) is consistent with the head-dominance estimate of Section 7: when the body is quantized too, the body term the estimate treats as small (∼2.4% under BF16/FP16) grows until it dominates, and a head-scope repair can no longer reach it.

The two FP8 configurations establish different claims. Because the head-isolated comparison shares BF16 weights and a BF16 transformer body, the hidden state entering the head is identical across arms; this removes both the measured BF16/FP16 weight-storage gap and body-originated divergence by construction. It therefore tests transfer of (i) the low-margin lm_head mechanism, (ii) the margin-gated repair, and (iii) format-dependent threshold calibration to FP8 head arithmetic. The end-to-end experiment quantises every linear layer and is the deployment-shaped stress test included here. Its +1pp result shows that the head-only method is largely ineffective once FP8 error accumulates through the body. We do not evaluate FP8 KV-cache quantisation or claim an end-to-end FP8 reproducibility solution.

XVI Execution-based correctness (HumanEval)

We run the OpenAI HumanEval execution sandbox on all four arms (BF16 baseline, FP16 baseline, BF16+C, FP16+C) using TinyLlama-1.1B (n=164, full benchmark).

The 68% token-level divergence produces zero execution-correctness flips: the divergent code completions are either both wrong or both right (in all observed cases, both wrong—TinyLlama's code accuracy is low). Intervention C introduces zero pass@1 degradation in this run, but the near-zero baseline accuracy makes this a weak quality-preservation test; the Qwen experiment below is the informative execution-based result.

A capable model: Qwen2.5-3B-Instruct. TinyLlama's near-zero code accuracy makes its "zero flips" result weak evidence: two arms that are both always wrong agree trivially. We therefore repeated the

Table 27: Execution-based pass@1 on HumanEval (TinyLlama-1.1B, n=164). Despite 68% token-level divergence between BF16 and FP16, execution correctness is identical across all configurations: zero correctness flips.

Config	pass@1 (%)	Correctness flips vs. BF16 baseline
BF16 baseline	0.6	—
FP16 baseline	0.6	0
BF16 + IntC	0.6	0
FP16 + IntC	0.6	0
Token-level divergence (BF16 vs FP16)		68%

experiment on Qwen2.5-3B-Instruct, which solves half of HumanEval, so that agreement is measured over a population where correctness genuinely varies (Table 28).

Table 28: Execution-based pass@1 on HumanEval (Qwen2.5-3B-Instruct, n=164, full benchmark). At a pass@1 of ∼50%, the two arms disagree on 73.8% of completions at the token level, yet agree on 97.6% of execution outcomes. Intervention C does not degrade pass@1.

Config	pass@1 (%)	Correctness flips vs. BF16 baseline
BF16 baseline	50.0	—
FP16 baseline	52.4	4 of 164 (2.4%)
BF16 + IntC	51.2	— (+1.2pp vs. BF16)
Token-level divergence (BF16 vs FP16)		73.8%
Execution-outcome agreement		97.6%

Three observations. First, the token-level/outcome-level gap is large and in the direction our discussion of practical adequacy predicts: 73.8% of completions differ somewhere in their token sequence, but 97.6% of them execute to the same verdict, so most cross-precision divergence on this task is not correctness-bearing. Second, unlike TinyLlama, the flips here are not zero: 4 of 164 problems flip, all in the same direction (FP16 correct, BF16 incorrect; exact binomial p=0.125, not significant at n=164). We report the direction rather than suppress it, but with 4 discordant pairs the data cannot support a claim that either format is systematically better on code, and the aggregate 2.4pp pass@1 difference is well inside the ±5pp non-inferiority margin used elsewhere in this paper. Third, Intervention C's pass@1 (51.2%) is above the BF16 baseline it repairs toward (50.0%), so the gated recomputation does not trade correctness for reproducibility on this benchmark either.

Across the two models, execution-outcome agreement greatly exceeds token-level agreement: 100% vs. 32% on TinyLlama and 97.6% vs. 26.2% on Qwen2.5-3B.

XVII Cross-task EAR on Qwen2.5-3B-Instruct

To verify that the phenomenon generalises beyond GSM8K, we report EAR on two additional code benchmarks for Qwen2.5-3B-Instruct.

Table 29: Cross-task EAR for Qwen2.5-3B-Instruct (n=100, τ=10^{-3}). Divergence rates are consistent across math and code tasks.

Task	EAR (%)	Divergence (%)	Median first-div step
GSM8K	25	75	—
HumanEval	37	63	step 83
MBPP	40	60	step 43

Both code benchmarks show 60–63% divergence on Qwen2.5-3B-Instruct, comparable to the 75% GSM8K divergence measured under the same protocol. The slightly higher EAR on code tasks is consistent with their shorter average generations and fewer observed low-margin steps per prompt.[4]

[4]The experiments in this appendix, together with the FP8, fidelity, and directional analyses, use a bare "Solve step by step" prompt format, under which Qwen2.5-3B GSM8K EAR is 25% (75% divergence); the main-text cross-model table uses the chat-template protocol, giving 18% EAR (82% divergence). Per-format numbers are therefore not directly comparable.