LOSSLESS, YET NOT IDENTICAL: EXECUTION-LEVEL NUMERICAL BIFURCATIONS IN SPECULATIVE DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Speculative decoding is lossless at the level of the decoding rule, but this guarantee does not require different finite-precision executions to produce identical decisions. We study this distinction in greedy speculative decoding and focus on an attribution problem: when speculative and autoregressive generation disagree, does the disagreement arise specifically under speculative verification, or is the same state already sensitive to the autoregressive execution path? We address this question with a three-way diagnostic that compares incremental autoregressive execution (A), full-prefix autoregressive execution (B), and speculative verification (C) at the same token prefix. We treat $A = B \neq C$ as verifier-specific disagreement relative to these two matched autoregressive controls. Among 555 reconstructed first-divergence events in Qwen3, 166 satisfy this criterion, while 389 are already autoregressive-path-sensitive. Across Qwen3, SmolLM2, and Pythia, all 1,850 reconstructed BF16 divergence events collapse to $A = B = C$ under backend-matched FP32 replay, which we use only as a diagnostic control. The observed bifurcations are strongly enriched near low-margin decisions, but their prevalence also varies with execution shape and backend. The results therefore do not support a universal margin threshold or block-size law. A targeted replication on A100/Ampere again identifies the same event class at both tested block sizes, and all 145 reconstructed events collapse under matched FP32 replay. The verifier-specific fraction is nevertheless lower than on RTX4090/Ada, showing that the event class persists across the two tested GPU architectures while its observed magnitude can differ. We further use Selective Canonical Re-verification as a mechanism-grounded intervention. Changing only the execution realization recovers most lost execution identity in most tested settings, while high-margin residuals expose the limits of margin-only selection and direct GPU timing quantifies the cost of the present implementation. These results separate two notions of exactness: losslessness constrains the decoder, whereas execution identity also depends on how that decoder is realized under finite precision.

1 INTRODUCTION

Speculative decoding accelerates autoregressive generation while preserving the target decoder under its mathematical specification Leviathan et al. (2023); Chen et al. (2023). In the greedy setting studied here, this guarantee constrains the next-token rule but not its finite-precision realization. The same model and token prefix can be evaluated through different cache constructions, tensor shapes, or execution backends, and these realizations need not produce identical logits Yuan et al. (2025); Chodavarapu & Xu (2026); Gond et al. (2026); Zhao et al. (2026). We therefore ask a separate execution-level question: **does mathematical losslessness imply execution-level identity?**

Execution-level numerical non-invariance in LLM inference is already known Yuan et al. (2025); Chodavarapu & Xu (2026); Koziev et al. (2026). The unresolved issue for speculative decoding is attribution. Speculative verification evaluates several draft tokens through a block-shaped target computation, whereas ordinary autoregressive decoding typically evaluates the target incrementally Leviathan et al. (2023); Chen et al. (2023); Miao et al. (2023). A speculative–autoregressive disagreement may therefore reflect numerical sensitivity already present across autoregressive exe-

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

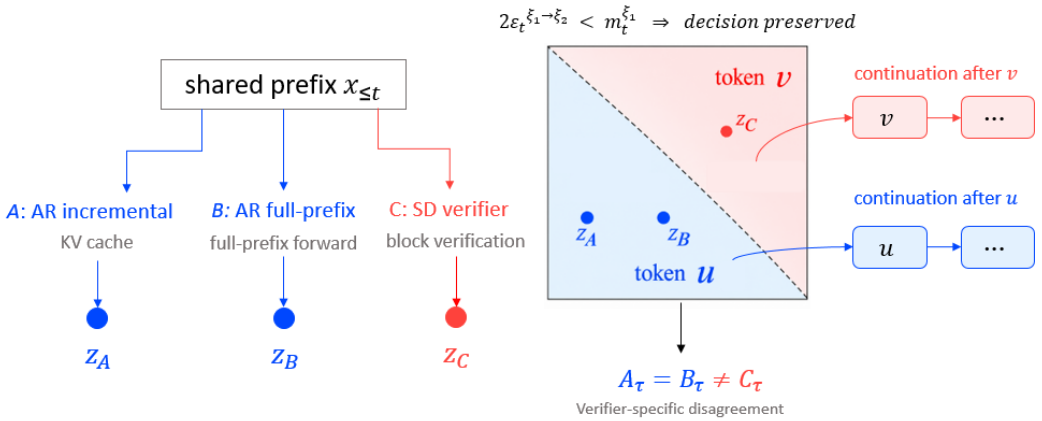


Figure 1: **Execution-level attribution under mathematically lossless greedy decoding.** At a shared prefix $x_{\leq t}$, we compare incremental autoregressive execution A, full-prefix autoregressive execution B, and speculative verification C. The schematic decision geometry illustrates how execution-induced numerical variation interacts with local decision margin, while $A_t = B_t \neq C_t$ denotes verifier-specific disagreement relative to the two matched autoregressive controls. Different next-token decisions can lead to different autoregressive continuations.

cution paths rather than disagreement specific to speculative verification within the tested comparison. A two-way comparison cannot distinguish these cases.

We address this ambiguity with a three-way diagnostic at the first differing token. At the same token prefix, we compare incremental autoregressive execution (A), full-prefix autoregressive execution (B), and speculative block verification (C). We treat $A = B \neq C$ as verifier-specific disagreement relative to these two matched autoregressive controls. This criterion does not identify a unique causal source; instead, it separates these events from states that are already sensitive to the autoregressive execution path. Figure 1 summarizes the matched execution views, the local decision geometry associated with $A_t = B_t \neq C_t$, and the resulting trajectory consequence. In Qwen3, 166 of 555 reconstructed first-divergence events satisfy $A = B \neq C$, while the remaining 389 are autoregressive-path-sensitive. Aggregate trajectory rates do not reveal this decomposition.

The reconstructed bifurcations concentrate near low-margin target decisions, but margin alone does not determine whether a token flips Chu et al. (2026); Song et al. (2026). Their prevalence also varies with speculative block size and attention backend, and the data do not support a universal larger-K instability law. The same event class recurs across Qwen3 Yang et al. (2025), SmolLM2 Allal et al. (2025), and Pythia Biderman et al. (2023), and a targeted A100/Ampere replication observes it again at a different magnitude. Backend-matched FP32 replay serves only as a diagnostic control.

We use Selective Canonical Re-verification (SCR) as a controlled intervention on execution realization. SCR reroutes selected verifier decisions through the canonical incremental target path while keeping the model, numerical precision, and decoding rule fixed. Across the tested settings, SCR recovers execution identity to varying degrees, while high-margin residuals expose the limits of margin-only selection. We observe no mitigation-induced new divergences across 19,680 held-out runs, but the present implementation incurs substantial wall-clock overhead. We therefore use SCR as a mechanism-grounded intervention rather than an optimized inference system.

Our study makes three contributions:

- We introduce a three-way diagnostic that separates verifier-specific $A = B \neq C$ disagreement, relative to two matched autoregressive controls, from disagreement already present across autoregressive execution paths.
- We show that these bifurcations vary with numerical precision, target decision margin, execution shape, and backend. The same event class appears across three model families and is observed again in a targeted replication on a second GPU architecture, while its observed magnitude varies across models and execution stacks.

- We use SCR as a controlled intervention to test recovery of execution identity and characterize both the limits of margin-only selection and the measured runtime cost of the present implementation.

Together, these results separate two notions of exactness in greedy speculative decoding: mathematical losslessness constrains the decoder, whereas execution identity also depends on its finite-precision realization.

2 Related Work

2.1 Speculative Decoding and Algorithmic Exactness

Speculative decoding reduces serial target-model computation by proposing multiple tokens before target verification (Leviathan et al., 2023; Chen et al., 2023). Later methods improve proposal quality or verification efficiency through tree-structured verification, self-speculation, multi-token prediction, and learned feature-space drafting (Miao et al., 2023; Zhang et al., 2024; Cai et al., 2024; Li et al., 2024). Although these methods differ in implementation, they share an exactness objective under their respective decoding specifications. We study greedy decoding, where exactness can be evaluated directly at the realized next-token decision. Orthrus shows that a speculative decoding method designed to be lossless can nevertheless lose trajectory identity under reduced precision (Koziev et al., 2026). Our question begins after such disagreement occurs: can it be distinguished from sensitivity already present across autoregressive execution paths?

2.2 Numerical Reproducibility in LLM Inference

Mathematically equivalent LLM computations can produce different finite-precision outputs under changes in batch configuration, hardware, precision, cache construction, tensor shape, or execution schedule (Yuan et al., 2025; Chodavarapu & Xu, 2026; Zhao et al., 2026). Chodavarapu and Xu show that KV-cached and recomputed autoregressive executions can diverge deterministically under reduced precision (Chodavarapu & Xu, 2026). This autoregressive execution-path sensitivity directly confounds speculative–autoregressive comparisons. Numerical non-identity alone therefore does not establish whether speculative verification contributes a distinct disagreement.

2.3 Execution Control, Selective Verification, and Attribution

Several systems reduce numerical variation by constraining execution or selectively re-evaluating vulnerable decisions. LLM-42 combines fixed-shape replay with rollback for deterministic inference, while CoRun controls execution shape to reduce batch-dependent variation (Gond et al., 2026; Zhao et al., 2026). MarginGate uses low target-logit margins to identify batch-induced token flips and selectively invokes additional verification (Chu et al., 2026), whereas MARS uses target-model margin to adapt speculative verification for efficiency (Song et al., 2026). These works establish canonical execution and margin-based selection as useful mechanisms, but neither mechanism is our primary contribution.

Our distinction is attribution. We compare incremental autoregressive execution (A), full-prefix autoregressive execution (B), and speculative verification (C) at the same decoding state, and treat $A = B \neq C$ as verifier-specific disagreement relative to the two matched autoregressive controls. Agreement between A and B rules out disagreement between those two autoregressive realizations at that state, but it does not identify a unique causal source or exclude every alternative execution-level explanation. The three-way diagnostic therefore complements prior work on numerical non-invariance and deterministic execution by asking whether a distinct disagreement remains under speculative verification after the matched autoregressive controls agree.

3 PROBLEM FORMULATION

3.1 DECODER EXACTNESS AND EXECUTION IDENTITY

We distinguish between exactness of the decoding rule and identity of its numerical execution. Let $x_{\le t}$ denote the token prefix at decoding step t, and let a target model with parameters θ define the greedy next-token rule

$$g_\theta(x_{\le t}) = \arg\max_{v \in \mathcal{V}} z_\theta(x_{\le t})_v. \tag{1}$$

Greedy speculative decoding changes how this decision is computed but is designed to preserve the same rule Leviathan et al. (2023); Chen et al. (2023). Mathematical losslessness therefore requires

$$g_\theta^{\mathrm{SD}}(x_{\le t}) = g_\theta^{\mathrm{AR}}(x_{\le t}) = g_\theta(x_{\le t}) \tag{2}$$

at every visited prefix.

This guarantee does not determine a unique finite-precision execution. We denote an execution configuration by $\xi = (e, p, b, h)$, where e is the execution path, p the numerical precision, b the backend, and h the hardware environment. Let $\tilde{\mathbf{z}}_\theta^{\xi}(x_{\le t})$ denote the logits produced under ξ, with realized decision

$$\tilde{g}_\theta^{\xi}(x_{\le t}) = \arg\max_{v \in \mathcal{V}} \tilde{z}_\theta^{\xi}(x_{\le t})_v. \tag{3}$$

For two valid executions ξ_1 and ξ_2 of the same mathematical decoder, *execution-level identity* holds at $x_{\le t}$ when

$$\tilde{g}_\theta^{\xi_1}(x_{\le t}) = \tilde{g}_\theta^{\xi_2}(x_{\le t}). \tag{4}$$

Execution-level identity concerns the realized token decision rather than bitwise equality of the underlying logits. Mathematical losslessness can therefore hold even when two valid executions produce different intermediate numerical values.

The local decision margin provides a simple stability criterion. Let $m_t^{\xi_1}$ denote the top-1 logit margin under ξ_1, and define

$$\varepsilon_t^{\xi_1 \to \xi_2} = \left\| \tilde{\mathbf{z}}_\theta^{\xi_2}(x_{\le t}) - \tilde{\mathbf{z}}_\theta^{\xi_1}(x_{\le t}) \right\|_\infty .$$

Then

$$2\varepsilon_t^{\xi_1 \to \xi_2} < m_t^{\xi_1} \quad \Longrightarrow \quad \tilde{g}_\theta^{\xi_1}(x_{\le t}) = \tilde{g}_\theta^{\xi_2}(x_{\le t}). \tag{5}$$

This condition is sufficient but not necessary. A small margin therefore indicates greater susceptibility to execution-induced perturbation, but it does not determine whether a disagreement occurs or how that disagreement should be attributed.

3.2 ATTRIBUTING THE FIRST DISAGREEMENT

A speculative–autoregressive disagreement is not sufficient for attribution because the same target state may already produce different decisions under alternative autoregressive execution paths. We therefore compare three executions of the same target model at the same token prefix while holding numerical precision, backend, and hardware fixed:

$$A_t = \tilde{g}_\theta^{\xi_A}(x_{\le t}), \qquad B_t = \tilde{g}_\theta^{\xi_B}(x_{\le t}), \qquad C_t = \tilde{g}_\theta^{\xi_C}(x_{\le t}), \tag{6}$$

where A denotes incremental autoregressive execution, B denotes full-prefix autoregressive execution, and C denotes speculative block verification.

We treat

$$A_t = B_t \neq C_t \tag{7}$$

as a *verifier-specific numerical bifurcation relative to the two matched autoregressive controls*. Agreement between A_t and B_t rules out disagreement between those two autoregressive realizations at that state. It does not identify a unique causal source or exclude every alternative execution-level explanation. By contrast, $A_t \neq B_t$ indicates that the state is already sensitive to the autoregressive execution path and therefore does not support the same attribution.

We apply this diagnostic separately to each matched autoregressive reference. For $R \in \{A, B\}$, let

$$\tau_R = \min\{t : R_t \neq C_t\}.$$

Table 1: **Execution-level non-identity and aggregate attribution ambiguity.** Target-only controls compare matched execution paths of the same target decoder. The two rightmost columns report aggregate trajectory disagreement between speculative verification and incremental autoregressive execution, and between incremental and full-prefix autoregressive execution.

	Target-only execution diagnostic		Aggregate trajectory disagreement	
Precision	**Path-state disagreement**	**Trajectory divergence**	**SD vs. inc. AR**	**Inc. AR vs. full-prefix AR**
FP32	0/118,784 (0%)	0%	0%	0%
FP16	43/118,784 (0.036%)	1.8–5.3%	3.7%	4.1%
BF16	273/118,784 (0.23%)	22–24%	23.6%	23.7%

Note. Target-only trajectory ranges summarize the tested execution-path comparisons and are not confidence intervals. FP32 zeros apply only to the evaluated configurations and do not imply a general guarantee of FP32 execution identity.

At each reconstructed τ_R, the three executions share the same preceding token history, so we evaluate A_{τ_R}, B_{τ_R}, and C_{τ_R} at the same decoding state. We classify the event as verifier-specific when $A_{\tau_R} = B_{\tau_R} \neq C_{\tau_R}$ and as autoregressive-path-sensitive when $A_{\tau_R} \neq B_{\tau_R}$. For brevity, we write τ when the reference is clear from context.

The attribution at τ is local to a single next-token decision. Once different tokens are selected, the resulting prefixes can diverge and subsequent autoregressive states can follow different trajectories. We therefore distinguish the *local numerical bifurcation* at τ from the *trajectory divergence* that may follow.

4 Revealing and Attributing Execution-Level Bifurcations

4.1 From State-Level Non-Identity to Trajectory Divergence

We first compare matched executions of the same target decoder using incremental, full-prefix, and block-forward execution paths while holding the model, token prefix, greedy decoding rule, numerical precision, and backend fixed within each comparison. Across 464 prompts, this protocol evaluates 29,696 decoding states per precision. These matched execution-path comparisons yield 118,784 path-state comparisons per precision, or 356,352 in total across FP32, FP16, and BF16.

The tested FP32 configurations produce no disagreement. FP16 and BF16 produce 43 (0.036%) and 273 (0.23%) path-state disagreements, respectively. These sparse path-state disagreements occur across 1.8–5.3% of FP16 trajectories and approximately 22–24% of BF16 trajectories, while the tested FP32 configurations show no trajectory divergence. Path-state and trajectory-level rates use different denominators and should not be interpreted as an amplification factor. Table 1 summarizes the results. Experimental configurations and evaluation populations are summarized in Appendix A.

4.2 Aggregate Rates Do Not Resolve Attribution

Aggregate trajectory rates do not identify the execution source of a disagreement. Under FP16, speculative–autoregressive trajectory divergence is 3.7%, compared with 4.1% between the two autoregressive execution paths. Under BF16, the corresponding rates are 23.6% and 23.7%. The tested configurations therefore show no simple aggregate increase in trajectory divergence under speculative decoding. Similar aggregate rates can nevertheless contain different first-disagreement classes. The condition $A_\tau \neq B_\tau$ indicates that the first differing state is already sensitive to the autoregressive execution path, whereas $A_\tau = B_\tau \neq C_\tau$ is verifier-specific relative to the two matched autoregressive controls. Aggregate frequency alone therefore cannot resolve attribution.

4.3 First-Divergence Attribution

We apply the diagnostic from Section 3.2 to the reconstructed first disagreement in the Qwen3 attribution analysis. Table 2 summarizes the resulting attribution classes.

Table 2: **First-divergence attribution in Qwen3.** Verifier-specific events satisfy $A_\tau = B_\tau \neq C_\tau$, whereas autoregressive-path-sensitive events satisfy $A_\tau \neq B_\tau$.

Attribution class	Count	Fraction
Verifier-specific	166	29.9%
AR-path-sensitive	389	70.1%
Total	555	100.0%

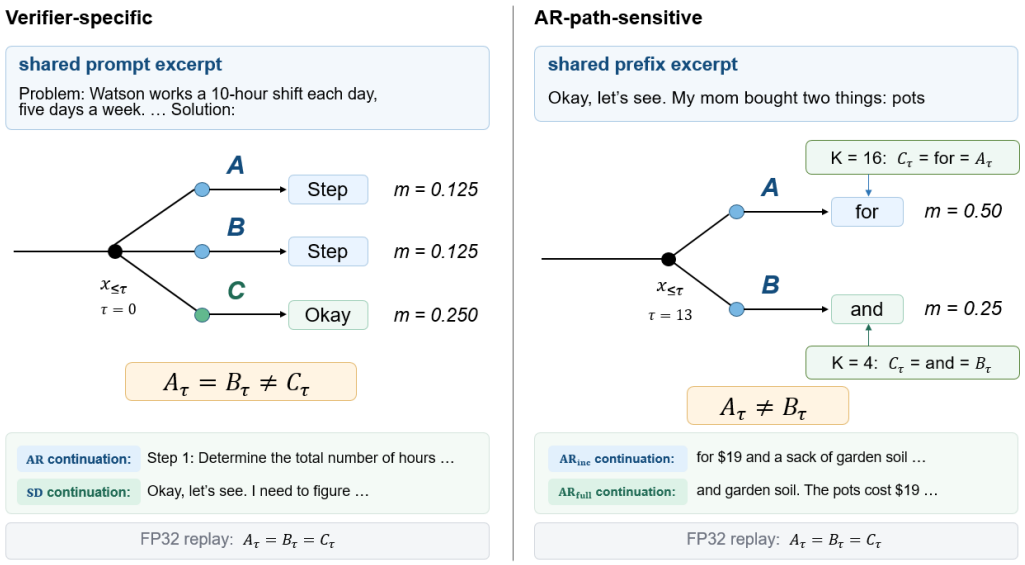


Figure 2: **Illustrative reconstructed examples of first-divergence attribution.** In the verifier-specific example, incremental and full-prefix autoregressive execution select `Step`, while speculative verification selects `Okay` at the same reconstructed state, yielding $A_\tau = B_\tau \neq C_\tau$. In the autoregressive-path-sensitive example, the two autoregressive executions already select different next tokens at the same state. Speculative verification aligns with B_τ for $K = 4$ and with A_τ for $K = 16$ at the illustrated first-divergence position. Matched FP32 replay collapses each illustrated event to a single target decision and is shown only as a diagnostic control, not as semantic ground truth.

The two-way speculative–autoregressive comparison therefore combines two distinct attribution classes: verifier-specific events and states that are already sensitive to the autoregressive execution path. First-divergence reconstruction separates these classes without assigning a unique causal source to the verifier-specific events. Figure 2 makes this distinction concrete using two illustrative reconstructed events from the targeted Qwen3 A100 analysis.

4.4 Finite-Precision Structure of the Bifurcations

Under backend-matched FP32 replay, all 555 reconstructed reduced-precision event states satisfy $A_\tau = B_\tau = C_\tau$. We use this replay only as a diagnostic control consistent with a finite-precision origin in the tested replay setting. The result does not establish FP32 as semantic ground truth or guarantee deterministic execution. Verifier-specific events are also enriched near low-margin target decisions, consistent with the susceptibility implied by Eq. 5. This association is not deterministic: low-margin states need not produce a token flip, and margin alone does not determine attribution.

Execution shape provides additional structure. The association with speculative block size is stronger under SDPA and flatter under eager execution, so the results do not support a universal larger-K instability law. A complementary state-level hazard analysis shows that block size remains associated with verifier-specific disagreement after adjustment for the other measured covariates. In an L2-regularized logistic model over verifier states up to and including the first speculative–

Table 3: **Cross-family recurrence with model-dependent magnitude.** First-divergence events are reconstructed on a separate RTX4090 holdout shared across the three model families. Verifier-specific events satisfy $A_\tau = B_\tau \neq C_\tau$. The final column reports the verifier-specific fraction for the SDPA speculative-versus-incremental-AR comparison as the speculative block size changes from $K = 4$ to $K = 16$.

Model	Reconstructed	Verifier-specific	Fraction	SDPA: $K = 4 \rightarrow 16$
Qwen3	451	218	48.3%	15.8% → 50.9%
SmolLM2	360	109	30.3%	1.8% → 15.5%
Pythia	1,039	488	47.0%	10.2% → 47.3%
Total	**1,850**	**815**	**44.1%**	–

Note. The holdout uses HumanEval 100 and GSM8K 100 with BF16/FP32 replay, $K \in \{4, 16\}$, and SDPA/eager execution. Eager execution shows a flatter block-size response. All 1,850 reconstructed BF16 events collapse under matched FP32 replay, which is used only as a diagnostic control.

incremental-AR divergence, a one-standard-deviation increase in K (approximately five tokens over the tested range) is associated with higher odds of $A_t = B_t \neq C_t$ (OR $= 2.34$, 95% prompt-bootstrap CI $[1.65, 3.23]$). We next test which aspects of this event structure recur across model families and hardware stacks. Appendix B provides reconstruction accounting and statistical details.

5 Recurrence Across Models and Hardware

5.1 Cross-Family Recurrence and Model-Dependent Structure

We next evaluate the first-divergence diagnostic on a separate RTX4090 holdout shared across Qwen3, SmolLM2, and Pythia. We reconstruct 451, 360, and 1,039 events, including 218, 109, and 488 verifier-specific cases, respectively. The event class appears in all three tested model families, but its prevalence differs across them. All 1,850 reconstructed BF16 events collapse to $A_\tau = B_\tau = C_\tau$ under backend-matched FP32 replay, which we use only as a diagnostic control. Table 3 summarizes the cross-family results.

The execution-dependent structure also recurs, but its magnitude remains model dependent. For the SDPA speculative-versus-incremental-AR comparison, the verifier-specific fraction changes from 15.8% to 50.9% for Qwen3, from 1.8% to 15.5% for SmolLM2, and from 10.2% to 47.3% for Pythia as K increases from 4 to 16. Eager execution shows a flatter response. Low-margin enrichment also appears across the three families, although the SmolLM2 pattern is not strictly monotonic. These results support recurring execution-dependent structure rather than a universal block-size law, margin threshold, or common effect magnitude.

5.2 Targeted Cross-Hardware Replication

We replicate the Qwen3 diagnostic on A100/Ampere using the same SDPA configuration with BF16, matched FP32 replay, $K \in \{4, 16\}$, and 200 prompts. We reconstruct 145 first-divergence events, including 26 verifier-specific and 119 autoregressive-path-sensitive cases. All 145 events collapse to $A_\tau = B_\tau = C_\tau$ under matched FP32 replay. The verifier-specific fraction is 17.9% on A100, compared with 42.5% in the condition-matched RTX4090 comparison rather than the aggregate Qwen3 holdout. The event class therefore appears on both tested GPU architectures, while its observed prevalence differs across the two execution stacks. Appendix C provides additional cross-family and cross-hardware breakdowns.

6 Recovering Execution Identity

6.1 Selective Canonical Re-verification

We use *Selective Canonical Re-verification* (SCR) as a controlled intervention to test whether changing the execution realization can recover execution identity. At verifier state t, SCR routes triggered

decisions through incremental autoregressive execution while leaving the remaining speculative decisions unchanged:

$$\tilde{g}_\theta^{\mathrm{SCR}}(x_{\le t}) = \begin{cases} A_t, & m_t^C \le \gamma, \\ C_t, & m_t^C > \gamma. \end{cases} \tag{8}$$

The target model, numerical precision, and decoding rule remain fixed; only the execution path changes at triggered states. Incremental BF16 execution serves as the operational reference, while FP32 remains a diagnostic control rather than the semantic target of SCR. Because margin profiles differ across model–backend settings, we calibrate γ separately for each family–backend pair on the disjoint 200-prompt calibration set rather than impose a global threshold.

6.2 Recovery and Selector Limitations

For the $q = 0.10$ budget policy, the resulting held-out trigger rates are approximately 12% of active states. SCR recovers 100% of divergent trajectories for Qwen3 and SmolLM2, 95.8% for Pythia/SDPA, and 54.3% for Pythia/eager (Figure 3a). Across 19,680 clean held-out runs spanning 216 policy–condition cells, we observe zero mitigation-induced new divergences relative to the incremental reference. The always-canonical control also yields zero divergence under the tested conditions. Recovery is therefore strong in several settings but remains model–backend dependent.

The Pythia/eager residuals expose the limitation of margin-only selection. In a 100-event residual audit, 98 events have recorded verifier margins, and all 98 exceed the $q = 0.10$ threshold of 0.5; two margin records are missing. The recorded margins range from 0.5625 to 3.25, placing the observed residuals outside the low-margin region targeted by the selector (Figure 3b). These results separate intervention efficacy from selector completeness: the always-canonical control eliminates divergence under the tested conditions, whereas margin-only selection misses states that later diverge.

6.3 Measured Runtime Cost

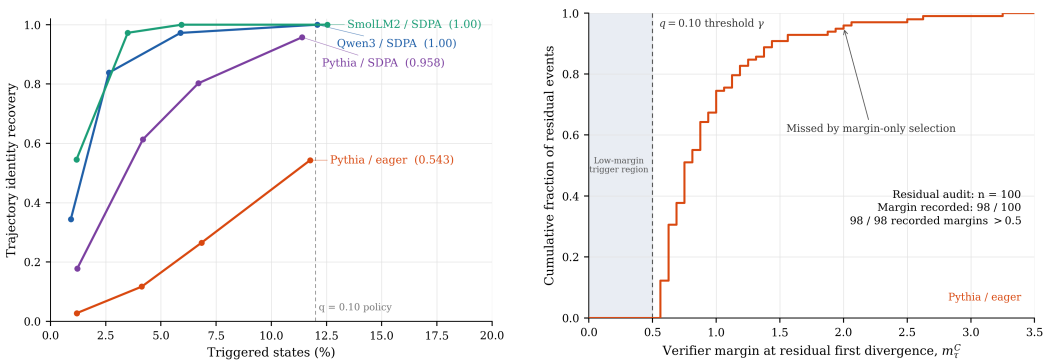
SCR is selective in trigger frequency but not in measured runtime cost. In direct A100 timing at $K = 4$, trigger rates are 12.56%, 13.25%, and 13.04% for Qwen3/SDPA, Pythia/SDPA, and Pythia/eager, respectively, while wall-clock overhead relative to vanilla speculative decoding is 75.1%, 76.8%, and 56.7% (Figure 3c). The Pythia timing measurements use the 50-prompt subset. Trigger frequency is therefore not a proxy for runtime cost in this implementation, where canonical re-verification and subsequent state synchronization remain expensive. Neither vanilla speculative decoding nor SCR exceeds autoregressive throughput under the tested timing conditions. We therefore use SCR as a mechanism-grounded intervention rather than an efficiency result. Appendix D provides calibration, recovery, residual, and runtime details.

7 Discussion and Limitations

Our results distinguish decoder-level exactness from execution-level identity. Greedy speculative decoding can preserve the mathematical next-token rule while valid finite-precision realizations select different tokens at the same prefix. The three-way diagnostic shows why speculative–autoregressive disagreement alone is insufficient for attribution: first-divergence reconstruction separates verifier-specific $A_\tau = B_\tau \neq C_\tau$ events from states already sensitive to the autoregressive execution path. Mathematical losslessness therefore constrains the decoder-level rule, whereas execution identity also depends on its numerical realization.

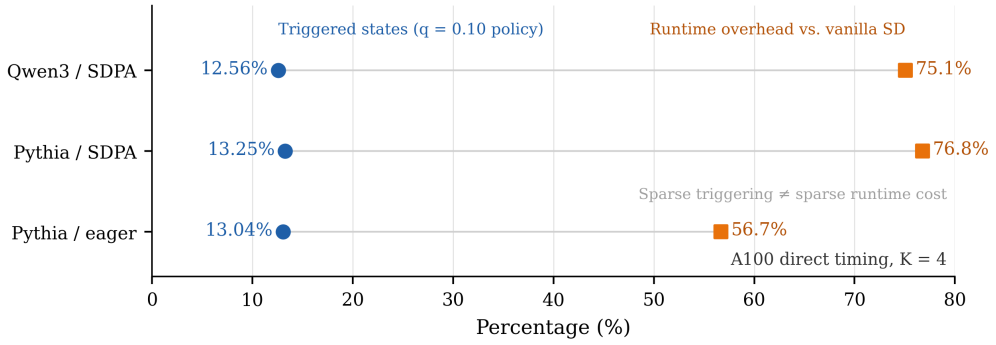
The attribution remains intentionally limited. Agreement between the two matched autoregressive controls does not identify a unique causal source or exclude alternative execution-level explanations. The observed relationships with decision margin, speculative block size, and backend describe structure rather than universal causal laws. Low margin indicates greater susceptibility but does not determine whether a token flips or how the disagreement should be attributed, and the block-size response varies across backends. Backend-matched FP32 replay is only a diagnostic control; it does not establish semantic ground truth or guarantee deterministic execution.

The empirical scope is also limited. We study Qwen3, SmolLM2, and Pythia on HumanEval and GSM8K under greedy decoding, with primary experiments on RTX4090/Ada and a targeted Qwen3



(a) Recovery vs. intervention rate.

(b) Residual selector failure.



(c) Triggered states vs. measured runtime overhead.

Figure 3: **SCR recovery, selector limitations, and measured runtime cost. (a)** Trajectory identity recovery as more verifier states are routed through incremental execution; the $q = 0.10$ policies produce held-out trigger rates near 12%. **(b)** Pythia/eager residual first-divergence margins. Among 100 audited events, 98 have recorded margins and all 98 exceed the $q = 0.10$ threshold of 0.5; two margin records are missing. **(c)** Direct A100 timing at $K = 4$. Trigger rates near 13% correspond to 56.7–76.8% wall-clock overhead relative to vanilla speculative decoding in the tested settings.

replication on A100/Ampere. The verifier-specific event class appears across the tested model families and both GPU architectures, but its prevalence is not quantitatively invariant. SCR is a controlled intervention rather than a universal remedy: margin-only selection misses high-margin residuals in Pythia/eager, and the current implementation incurs substantial wall-clock overhead despite sparse triggering. The present evidence does not establish hardware invariance, a universal selector, production efficiency, or generalization to stochastic decoding.

8 Conclusion

Speculative decoding can be mathematically lossless without being execution-identical. Losslessness constrains the decoder-level rule, while finite-precision execution can still change the discrete decision realized at a given prefix. Our three-way diagnostic separates verifier-specific $A_\tau = B_\tau \neq C_\tau$ bifurcations from disagreements that are already present across autoregressive execution paths. First-divergence analysis further reveals structured variation with decision margin, execution shape, backend, and model family across the tested settings. Selective canonical re-verification shows that changing the execution realization can recover identity when vulnerable states are identified, while high-margin residuals expose the limits of margin-only selection. These results establish execution identity as distinct from mathematical losslessness: losslessness specifies the decoder-level rule, whereas the realized trajectory also depends on the finite-precision execution path through which that rule is evaluated.

AI USE STATEMENT

We used generative AI tools to assist with writing and language polishing and to support retrieval and discovery of potentially relevant literature. All AI-assisted outputs were reviewed by the authors, and literature suggestions were independently checked before citation. The authors take full responsibility for the final manuscript, including its technical content, claims, citations, and presentation.

REPRODUCIBILITY STATEMENT

We provide the experimental configurations, model and dataset settings, execution backends, numerical precisions, and evaluation populations in Appendix A. Appendix B describes the first-divergence reconstruction procedure and statistical analysis, Appendix C provides the cross-family and cross-hardware evaluation details, and Appendix D documents SCR calibration, residual analysis, and runtime measurement. These materials specify the protocols and evaluation populations used for the reported results.

REFERENCES

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International conference on machine learning*, pp. 2397–2430. PMLR, 2023.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*, 2024.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Ranjith Chodavarapu and Lei Xu. The illusion of equivalence: Systematic fp16 divergence in kv-cached autoregressive inference. *arXiv preprint arXiv:2604.15409*, 2026.

Kexin Chu, Yang Zhou, and Wei Zhang. Margingate: Sparse margin-triggered verification for batch-invariant llm inference. *arXiv preprint arXiv:2605.30218*, 2026.

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.

Ilya Koziev, Leonid Sinev, and Ivan Oseledets. How lossless is lossless speculative decoding? the role of numerical precision in orthrus. *arXiv preprint arXiv:2609.15504*, 2026.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International conference on machine learning*, pp. 19274–19286. PMLR, 2023.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, et al. Specinfer: Accelerating generative large language model serving with tree-based speculative inference and verification. *arXiv preprint arXiv:2305.09781*, 2023.

Jingwei Song, Xinyu Wang, Hanbin Wang, Xiaoxuan Lei, Tianyu Shi, Shixin Han, Eric Yang, Xiao-Wen Chang, and Lynn Ai. Mars: Unleashing the power of speculative decoding via margin-aware verification. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 8652–8663, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Jiayi Yuan, Hao Li, Xinheng Ding, Wenya Xie, Yu-Jhe Li, Wentian Zhao, Kun Wan, Jing Shi, Xia Hu, and Zirui Liu. Give me fp32 or give me death? challenges and solutions for reproducible reasoning. *arXiv e-prints*, pp. arXiv–2506, 2025.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. Draft& verify: Lossless large language model acceleration via self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11263–11282, 2024.

Shiju Zhao, Jiacheng Yang, Qihang Chen, Junhao Hu, Jiaqi Zheng, Guihai Chen, and Xusheng Chen. Corun: Padding is simple and efficient for deterministic llm inference. *arXiv preprint arXiv:2608.14376*, 2026.

A Experimental Setup and Evaluation Populations

A.1 Experimental Configuration

We evaluate three target–draft model pairs: Qwen3-4B with Qwen3-0.6B, SmolLM2-1.7B with SmolLM2-135M, and Pythia-1.4B with Pythia-160M. All model revisions are pinned in the experimental configuration. The primary RTX4090 experiments use Python 3.11.10, PyTorch 2.4.1 with CUDA 12.4, and Transformers 4.55.2, while the A100 replication uses the same library versions. Unless stated otherwise, decoding is greedy with sampling disabled, batch size one, and a maximum of 64 generated tokens. TF32 is disabled, and the implementation uses neither `torch.compile` nor autocast. We follow a fresh-cache execution protocol and do not reuse KV caches across speculative blocks or prompts.

The execution backend varies across experimental stages. The initial target-only and Qwen3 attribution analyses use SDPA. The cross-family and SCR experiments evaluate both SDPA and eager execution, while the targeted A100 replication uses SDPA only. The initial Qwen3 analyses use speculative block sizes $K \in \{4, 8, 16\}$, whereas the cross-family, cross-hardware, and SCR evaluations use $K \in \{4, 16\}$.

A.2 Evaluation Populations

Table 4 summarizes the distinct experimental populations used in the study. These populations are not interchangeable. The Qwen3 attribution analysis, the separate RTX4090 cross-family holdout, and the A100 replication differ in prompt population, precision mix, backend coverage, or hardware. Event counts are indexed by prompt, speculative block size, and comparison pair; Appendix B provides the corresponding unique-state accounting.

The 200-prompt cross-family set also serves as the SCR calibration population and as the prompt set for the targeted A100 replication. The SCR held-out set contains 64 HumanEval and 136 GSM8K prompts and has zero prompt overlap with the 200-prompt calibration set; the evaluation pipeline explicitly checks this separation. The held-out prompts were used in earlier Qwen-only experimental phases, so the disjointness claim applies specifically to the 200-prompt SCR calibration set and the 200-prompt held-out SCR evaluation set.

Table 4: **Experimental populations used in the main analyses.** HE denotes HumanEval. Event counts are event-weighted unless otherwise stated.

Analysis	Model(s)	Prompts	Precision / backend	K	Population
Target-only diagnostic	Qwen3	164 HE + 300 GSM8K	FP32 / FP16 / BF16; SDPA	4, 8, 16	29,696 decoding states; 118,784 path comparisons per precision
Qwen3 attribution	Qwen3	164 HE + 300 GSM8K	BF16 + FP16; SDPA	4, 8, 16	555 reconstructed events
Cross-family holdout	Qwen3 / SmolLM2 / Pythia	100 HE + 100 GSM8K	BF16; SDPA + eager	4, 16	1,850 reconstructed events
RTX4090 comparator	Qwen3	Same 200 prompts	BF16; SDPA	4, 16	179 reconstructed events
A100 replication	Qwen3	Same 200 prompts	BF16; SDPA	4, 16	145 reconstructed events
SCR calibration	All three families	100 HE + 100 GSM8K	BF16; SDPA + eager	4, 16	147,219 active verifier states
SCR held-out	All three families	64 HE + 136 GSM8K	BF16; SDPA + eager	4, 16	200 disjoint held-out prompts

Table 5: **First-divergence reconstruction accounting.** Events are indexed by prompt, block size, and comparison pair. Unique-state counts merge repeated observations of the same decoding state.

Population	Events	Unique states	VS events	VS unique	FP32 agree
Qwen3 attribution	555	314	166	130	555/555
Cross-family RTX4090	1,850	1,441	815	706	1,850/1,850
A100 replication	145	96	26	24	145/145

B First-Divergence Reconstruction and Statistical Analysis

B.1 First-Divergence Reconstruction

For each prompt, speculative block size, and autoregressive reference, we identify the first token position at which speculative decoding and the reference trajectory disagree and denote it by τ. The compared executions share the same committed prefix up to τ, so all three target executions can be evaluated at the same decoding state.

At the reconstructed state, A_τ is obtained from incremental autoregressive execution, B_τ from a full-prefix forward pass without cache reuse, and C_τ by re-evaluating the corresponding speculative verification block. If the first divergence occurs at a bonus position, we use the associated bonus-verification row. We classify an event as verifier-specific when

$$A_\tau = B_\tau \neq C_\tau,$$

and as autoregressive-path-sensitive when

$$A_\tau \neq B_\tau.$$

The reconstruction recomputes the target decisions directly rather than relying on stored argmax outputs.

For matched FP32 replay, we preserve the prompt, shared prefix, speculative block, draft tokens, backend, and block size while changing only numerical precision. We use FP32 replay only as a diagnostic control and do not treat it as a semantic reference.

Table 5 summarizes the reconstruction accounting. Events are indexed by prompt, block size, and speculative–autoregressive comparison pair, so multiple events can correspond to the same underlying decoding state. The event-weighted counts reported in the main text therefore differ from the corresponding unique-state counts.

Table 6: **Cross-family first-divergence attribution on RTX4090.** Each entry reports verifier-specific events over reconstructed events, followed by the corresponding fraction. Pooled values combine SDPA and eager execution.

Model	SDPA	Eager	Pooled
Qwen3	76/179 (42.5%)	142/272 (52.2%)	218/451 (48.3%)
SmolLM2	20/132 (15.2%)	89/228 (39.0%)	109/360 (30.3%)
Pythia	142/328 (43.3%)	346/711 (48.7%)	488/1,039 (47.0%)

All candidate divergences in these analyses are successfully reconstructed. The Qwen3 attribution analysis contains 555 reconstructed events, the separate cross-family RTX4090 analysis contains 1,850, and the A100 replication contains 145. No reconstruction failures occur in any of these populations.

B.2 State-Level Hazard Analysis

We use a complementary logistic regression to characterize the association between execution variables and verifier-specific disagreement at the state level. The analysis contains 266,438 verifier states from 464 prompts. For each trajectory, we retain states up to and including the first speculative–incremental-AR divergence. The binary outcome is

$$H_{\mathrm{SD}} = \mathbb{1}[A_t = B_t \neq C_t]\,,$$

with 85 positive states in the analyzed population.

The predictors are $\log_{10}(m_t^A + 10^{-8})$, speculative block size K, within-block position, normalized sequence position, committed-prefix length, numerical precision, dataset, and execution role. Execution role distinguishes accepted-draft verification, first-rejection correction, and bonus positions. We dummy-code categorical variables and standardize all regression columns before fitting. We then fit an L2-regularized logistic regression with $C = 1$.

The reported block-size association is defined per one standard deviation of K. In this population, K has mean 9.43 and standard deviation 5.00, so one standard deviation corresponds to approximately five tokens over the tested range. The fitted odds ratio is

$$\mathrm{OR} = 2.34,$$

with a 95% prompt-level bootstrap confidence interval of

$$[1.65,\ 3.23].$$

We obtain this interval from 500 bootstrap refits that resample prompts rather than individual states. This analysis is descriptive: the coefficient quantifies an adjusted association and does not establish a causal effect of block size.

C Cross-Family and Cross-Hardware Analysis

C.1 Cross-Family Attribution

We evaluate whether the first-divergence attribution structure recurs across Qwen3, SmolLM2, and Pythia on the separate 200-prompt RTX4090 holdout. All runs use BF16 with $K \in \{4, 16\}$ and evaluate both SDPA and eager execution. Table 6 reports the event-weighted attribution results for each model family and backend.

Across the three model families, the holdout contains 1,850 reconstructed first-divergence events, including 815 verifier-specific events. All 1,850 reconstructed events collapse to a single target decision under matched FP32 replay. The attribution classes therefore recur across the tested model families and execution backends, while their observed frequencies vary across settings.

The block-size comparisons reported in the main text use only the SDPA speculative-versus-incremental-AR comparison. Under this comparison, the verifier-specific fraction changes from

Table 7: **Condition-matched Qwen3 comparison across hardware.** Results are event-weighted and use SDPA, BF16, the same 200 prompts, and $K \in \{4, 16\}$.

Hardware	Events	VS events	VS fraction
RTX4090	179	76	42.5%
A100	145	26	17.9%

15.8% to 50.9% for Qwen3, from 1.8% to 15.5% for SmolLM2, and from 10.2% to 47.3% for Pythia as K increases from 4 to 16. These percentages use different denominators from the pooled family-level fractions in Table 6 and should therefore be interpreted separately.

C.2 Cross-Hardware Replication

We perform a targeted Qwen3 replication on an A100 using the same 200 prompts as the RTX4090 cross-family evaluation. The matched comparison preserves the model checkpoints, SDPA backend, BF16 precision, greedy decoding configuration, and $K \in \{4, 16\}$. The hardware environment and driver differ between the two runs. Table 7 reports the corresponding event-weighted comparison.

The verifier-specific event class appears on both tested GPU architectures, but its observed prevalence differs between the matched RTX4090 and A100 runs. The A100 analysis contains 145 reconstructed first-divergence events, including 26 verifier-specific and 119 autoregressive-path-sensitive events. All 145 reconstructed events collapse to a single target decision under matched FP32 replay. We therefore use this experiment to test recurrence across hardware environments rather than invariance of event rates.

D SCR Calibration, Recovery, Residual Analysis, and Runtime

D.1 Calibration and Trigger Rates

We calibrate SCR thresholds on BF16 verifier states from the 200-prompt calibration set, pooling $K \in \{4, 16\}$ within each model–backend setting. We retain only active verifier positions, exclude stale positions after the first rejection, and exclude bonus positions from calibration. Thresholds are calibrated separately for each model family and backend.

For a budget level q, we sort the calibration margins and set the threshold γ to the order statistic at index $\lceil qn \rceil - 1$. A state is triggered when its verifier margin satisfies $m_t^C \leq \gamma$. Under the $q = 0.10$ budget policy, the resulting thresholds are $\gamma = 0.875$ for Qwen3, $\gamma = 0.625$ for SmolLM2, and $\gamma = 0.5$ for Pythia under both tested backends. Tied quantized margins and the non-strict threshold rule yield realized calibration trigger rates between 10.8% and 11.6%.

The held-out SCR evaluation uses a disjoint 200-prompt set. Its pooled trigger rates range from approximately 11.4% to 12.5% across model–backend settings. Tied margins and the different dataset mixture between calibration and held-out evaluation partly explain the difference from the nominal 10% budget. We report $q = 0.10$ as the headline budget policy. This policy was selected from the evaluated budget sweep after examining held-out recovery and is therefore not treated as a prespecified primary policy.

D.2 Recovery and Residual Divergences

For each dataset–K cell, we define relative recovery as

$$1 - \frac{r_{\mathrm{SCR}}}{r_{\mathrm{baseline}}},$$

where r_{baseline} and r_{SCR} denote the trajectory-level divergence rates before and after SCR. We report the unweighted mean across the four dataset–K cells. Table 8 summarizes the resulting recovery for each model–backend setting.

The incomplete recovery for Pythia/eager motivates a separate residual audit. We sample 100 residual trajectories from the 156 observed under this setting. All 100 residual events reproduce on

Table 8: **SCR recovery under the $q = 0.10$ budget policy.** Values are unweighted means of the four dataset–K cell-wise relative recoveries.

Model	SDPA	Eager
Qwen3	100.0%	100.0%
SmolLM2	100.0%	100.0%
Pythia	95.8%	54.3%

Table 9: **SCR runtime on A100 at $K = 4$.** Overhead is measured relative to vanilla speculative decoding under the same timing protocol.

Setting	Trigger rate	SCR overhead
Qwen3 / SDPA	12.56%	75.1%
Pythia / SDPA	13.25%	76.8%
Pythia / eager	13.04%	56.7%

regeneration and remain deterministic across three repeats. Verifier margins are recorded for 98 events; the remaining two occur at bonus positions for which the stored verifier-margin array has no corresponding entry. All 98 recorded margins exceed the calibration threshold $\gamma = 0.5$, with values ranging from 0.5625 to 3.25. These residuals show that a low-margin trigger does not capture every divergence-prone state.

Across the full held-out policy grid, we observe no mitigation-induced new divergences in 19,680 runs spanning 216 policy-condition cells. This grid includes baseline and always-canonical controls, leaving 168 nontrivial SCR cells. The always-canonical control also produces zero divergence from the incremental autoregressive reference across all 480 tested runs.

D.3 Runtime Evaluation

We measure runtime on an A100 using BF16, $K = 4$, and the same fresh-cache implementation used in the main experiments. The Qwen3 timing evaluation uses 100 prompts, while each Pythia evaluation uses a deterministic 50-prompt subset. Each method receives five warm-up prompts followed by five timed repeats with rotating method order. We synchronize CUDA before and after each timed generation call.

Table 9 reports the held-out trigger rate and SCR overhead relative to vanilla speculative decoding. We compute the overhead as

$$\frac{T_{\mathrm{SCR}}}{T_{\mathrm{SD}}} - 1.$$

Neither vanilla speculative decoding nor SCR exceeds incremental autoregressive throughput under this fresh-cache timing implementation. We therefore use SCR as a controlled intervention on execution realization rather than as an optimized inference method.