Latent Phase-Shift Rollback: Inference-Time Error Correction via Residual Stream Monitoring and KV-Cache Steering

Anonymous Authors[1]

Abstract

Large language models frequently commit unrecoverable reasoning errors mid-generation: once a wrong step is taken, subsequent tokens compound the mistake rather than correct it. We introduce **Latent Phase-Shift Rollback** (LPSR): at each generation step, we monitor the residual stream at a critical layer ℓ_{crit}, detect abrupt directional reversals (*phase shifts*) via a cosine-similarity + entropy dual gate, and respond by rolling back the KV-cache and injecting a pre-computed steering vector. No fine-tuning, gradient computation, or additional forward passes are required. LPSR achieves $\mathbf{44.0}\%$ on MATH-500 with an 8B model versus 28.8% for standard AR ($+15.2$ pp; McNemar $\chi^2 = 66.96$, $p < 10^{-15}$). Critically, prompted self-correction, the most natural inference-time baseline, scores only 19.8%, *below* standard AR; LPSR exceeds it by $+24.2$ pp ($\chi^2 = 89.4$, $p \approx 0$). LPSR also outperforms Best-of-16 ($+7.8$ pp) at $5.4\times$ lower token cost, and surpasses a standard 70B model (35.2%) with $8.75\times$ fewer parameters at $\sim 3\times$ the token budget. A 32-layer sweep reveals a novel **detection–correction dissociation**: error-detection AUC peaks at layer 14 (0.718) but task accuracy peaks at layer 16 (44.0% vs. 29.2%), demonstrating that optimal monitoring depth differs for detection and correction.

1. Introduction

The ability to perform multi-step reasoning remains a fundamental challenge for large language models. Under greedy decoding, Llama-3-8B fails on 71.2% of MATH-500 problems; a standard Llama-3-70B model fails on 64.8%, an improvement of only 6.4 pp despite $8.75\times$ more parameters. A key failure mode is *error propagation*: elementary mistakes such as sign errors, wrong formula applications, variable confusions, compound undetected over hundreds of subsequent tokens (Wei et al., 2022; Lightman et al., 2024). Existing remedies fall into two categories: *training-time* approaches (fine-tuning on process-supervised reward models (Lightman et al., 2024; Uesato et al., 2022)) and *inference-time* approaches (chain-of-thought prompting (Wei et al., 2022), self-consistency (Wang et al., 2023), Best-of-N sampling (Snell et al., 2025), or tree search (Yao et al., 2023)). Training-time methods are expensive and tied to a fixed model. Inference-time methods either ignore the model's internal state entirely (Best-of-N) or require many additional forward passes (tree search, self-consistency).

We ask: *Can we detect a reasoning error while it is forming, before it has fully propagated, and correct the trajectory with minimal additional computation?*

The internal geometry of transformer residual streams offers an answer. Elhage et al. (2021) and Anthropic (2022) showed that the residual stream at middle layers encodes semantic content that is manipulable via linear steering vectors. Recent mechanistic interpretability work (Zou et al., 2024; Turner et al., 2023) demonstrates that these directions can be found unsupervised and generalise across tasks. Our observation is simpler: when a model is about to commit a reasoning error, the direction of the residual stream at layer $\ell_{\text{crit}} \approx L/2$ undergoes a sharp *phase shift*, a cosine similarity reversal between consecutive generation steps, that can be detected in real time.

Contributions. We make five contributions:

1. **LPSR algorithm.** A training-free inference method that monitors the residual stream, detects phase shifts via dual-gate authentication (cosine similarity + token entropy), and applies KV-cache rollback with steering vector injection (Section 3).
2. **Empirical validation.** LPSR achieves 44.0% on MATH-500, outperforming all 8B baselines and a standard 70B model at $\sim 3\times$ the token cost (Section 4).
3. **Layer dissociation finding.** A comprehensive 32-layer sweep reveals that error-*detection* AUC and task *accu-*

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

racy peak at different layers (14 vs. 16), a novel finding that informs optimal ℓ_{crit} selection (Section 5).

4. **Error characterisation.** Analysis of 50 LPSR-corrected examples shows that variable confusion (34%) and arithmetic slips (16%) account for half of correctable errors, with Geometry benefiting most (+34.2 pp; Section 5).
5. **Prompted self-correction characterisation.** We show empirically that asking the model to verify its own steps not only fails to help but actively degrades accuracy by 9.0 pp, providing systematic empirical evidence for why natural-language self-correction fails at this model scale (Huang et al., 2024) and why residual-stream monitoring is a more effective alternative.

2. Background and Related Work

Inference-time compute scaling. Snell et al. (2025) showed that inference-time compute can substitute for model scale on reasoning tasks. Best-of-N sampling and self-consistency (Wang et al., 2023), a majority-voted variant of Best-of-N, are the dominant approaches, but both require N independent forward passes and ignore the model's internal dynamics. Yao et al. (2023) and Besta et al. (2024) use tree structures to guide search, at greater computational cost. LPSR requires on average $\sim 3\times$ the token budget of greedy decoding, less than Best-of-N=16 (15.9$\times$ the token budget), and requires no tree structure.

Steering vectors and residual streams. Linear representation of semantic content in residual streams is well-established (Park et al., 2024; Zou et al., 2024; Turner et al., 2023). Zou et al. (2024) demonstrate that directions encoding high-level concepts can be extracted unsupervised and used to steer generation at inference time. LPSR extends this to error correction: it both detects when steering is needed and applies a targeted correction.

KV-cache manipulation. Speculative decoding (Leviathan et al., 2023) exploits KV-cache structure for speed whereas LPSR uses rollback for correctness. Yang et al. (2024) propose context-window trimming. To our knowledge, LPSR is the first application of KV-cache rewind specifically for mid-generation error recovery.

Process supervision and self-correction. Process reward models (Lightman et al., 2024) supervise intermediate steps but require labelled data and fine-tuning. Prompted self-correction (Madaan et al., 2023) asks the model to verify its own steps via an additional system prompt; our experiments show this baseline achieves only 19.8% on MATH-500 with Llama-3-8B, *lower* than standard AR (28.8%), consistent with Huang et al. (2024), who show that LLMs cannot reliably self-correct without external feedback.

Latent-space and continuous-token reasoning. Coconut (Hao et al., 2024) extends reasoning through latent continuous tokens, operating entirely in representation space across full generation. STIR-Static (a static-steering baseline we introduce in Section 4) applies a fixed steering vector without any detection mechanism. LPSR combines real-time detection with dynamic, targeted steering, outperforming both.

Concurrent scaling work. DeepSeek-R1 (Guo et al., 2025) and related "thinking" models achieve strong reasoning via extended chain-of-thought trained with reinforcement learning, a training-time approach requiring orders of magnitude more compute than LPSR. Shojaee et al. (2025) observe that apparent reasoning in large models may reflect shallow pattern matching ("the illusion of thinking"). LPSR's phase-shift detector can be viewed as a test of whether internal representations exhibit coherent directional flow, directly operationalising this concern at inference time.

3. Method: Latent Phase-Shift Rollback

3.1. Motivation: Phase Shifts as Error Precursors

Let $\{h_t^{(\ell)}\}_{t=1}^{T} \subset \mathbb{R}^d$ denote the hidden state at layer ℓ and generation step t. Define the *directional velocity*:

$$v_t^{(\ell)} = \frac{h_t^{(\ell)}}{\|h_t^{(\ell)}\|}, \qquad c_t^{(\ell)} = \langle v_t^{(\ell)},\, v_{t-1}^{(\ell)} \rangle. \tag{1}$$

We call $c_t^{(\ell_{\text{crit}})} < -\tau_\phi$ a *phase shift*: the representation at ℓ_{crit} is moving in the opposite direction to the previous step. Empirically, phase shifts at $\ell_{\text{crit}} = 16$ predict final answer incorrectness with AUC 0.652 on MATH-500 (layer sweep detailed in Section 5).

A single cosine gate is insufficient, token repetitions can cause low cosine similarity without error (empirically, 78% of low-entropy phase shifts occur during correct reasoning; see Appendix B). We therefore add a token-distribution entropy gate:

$$H_t = -\sum_j p_j \log p_j, \qquad p_j = \text{softmax}\Big(W_U h_t^{(\ell_{\text{crit}})}\Big)_j, \tag{2}$$

where W_U is the unembedding matrix. A phase shift is *authenticated* iff $c_t^{(\ell_{\text{crit}})} < -\tau_\phi$ *and* $H_t > \tau_H$; otherwise the token is emitted normally and generation continues. This dual-gate design reduces the false-positive rate to 22.0% while maintaining precision 0.784 and recall 0.267, a high-precision, low-recall design that accepts missed errors in exchange for confident corrections (Section 5.2).

3.2. Steering Vector Basis

We pre-compute a basis $\mathcal{V} = \{\delta_1, \ldots, \delta_K\} \subset \mathbb{R}^d$ of K=142 unit-norm steering vectors at layer ℓ_{crit} using the following procedure on a held-out calibration set:

1. Run the model on 1,000 MATH-500 training-split problems under standard AR.
2. For each problem where the answer is wrong, record the residual stream at the first phase-shift step $t^* = \min\{t : c_t^{(\ell_{\text{crit}})} < -\tau_\phi\}$.
3. Compute a *correction delta*: $\Delta_i = \tilde{h}_{t^*}^{(\ell_{\text{crit}})} - h_{t^*}^{(\ell_{\text{crit}})}$, where $h_{t^*}^{(\ell_{\text{crit}})}$ is the wrong-trajectory hidden state and $\tilde{h}_{t^*}^{(\ell_{\text{crit}})}$ is the corresponding state from a teacher-forced correct trajectory (i.e., decoded from the gold solution string).
4. Apply k-means clustering (k=256) on $\{\Delta_i\}$ and set δ_i to the ℓ_2-normalised cluster centroid.

At inference, when a phase shift is authenticated, the steering direction is selected as:

$$\delta^* = \arg\max_{\delta_i \in \mathcal{V}} \langle \delta_i,\, h_t^{(\ell_{\text{crit}})} \rangle, \tag{3}$$

i.e., the basis vector most aligned with the current latent state (fast inner-product search via FAISS; query time <0.1 ms). This selection is greedy-optimal within the span of $\mathcal{V}$ under a first-order Taylor approximation of the correction objective (Theorem B.4). The selected δ^* is then used in the KV-cache rollback and injection procedure described in Section 3.3.

3.3. KV-Cache Rollback and Injection

When a phase shift is authenticated at step t:

1. **Rollback.** Restore the KV-cache to state $t-1$, discarding the last generated token.
2. **Inject.** Modify the output of layer ℓ_{crit} for the re-decode by adding $\alpha \cdot \delta^*$, where α is an adaptive scale:

$$\alpha = \min\left(\alpha_{\max},\ \frac{|c_t^{(\ell_{\text{crit}})}|}{\tau_\phi} \cdot \alpha_{\max}\right), \tag{4}$$

   so that $\alpha = \alpha_{\max}$ for all authenticated shifts ($|c_t^{(\ell_{\text{crit}})}| \geq \tau_\phi$ is guaranteed by the gate condition) while remaining a well-defined continuous function of $|c_t^{(\ell_{\text{crit}})}|$ for analysis purposes (Appendix B).
3. **Re-decode.** Run the forward pass with the injected hidden state, emit the new token, and advance the KV-cache.

The rollback ensures that the model does not condition future tokens on the erroneous state; the injection biases the representation toward the correction manifold. The full procedure is given in Algorithm 1.

Algorithm 1 Latent Phase-Shift Rollback (LPSR)

Require: Model $\mathcal{M}$, tokenizer, prompt x, basis $\mathcal{V}$, parameters $\ell_{\text{crit}}, \tau_\phi, \tau_H, \alpha_{\max}$, max tokens T

Register forward hook at layer ℓ_{crit} to capture $h_t^{(\ell_{\text{crit}})}$
Encode prompt: kv $\leftarrow \mathcal{M}$.encode(x); $v_{\text{prev}} \leftarrow \mathbf{0}$ {$\mathbf{0}$ ensures no detection at $t = 1$}
for $t = 1, \ldots, T$ **do**
  $(\hat{y}_t,\ h_t,\ \text{kv}') \leftarrow \mathcal{M}$.step(kv)
  $c_t \leftarrow \langle v_t,\ v_{\text{prev}} \rangle$; $H_t \leftarrow \text{entropy}(h_t)$
  if $c_t < -\tau_\phi$ **and** $H_t > \tau_H$ **then** {Phase shift authenticated}
    $\delta^* \leftarrow \arg\max_{\delta_i \in \mathcal{V}} \langle \delta_i,\ h_t \rangle$ {FAISS inner-product query}
    $\alpha \leftarrow \min(\alpha_{\max},\ |c_t|/\tau_\phi \cdot \alpha_{\max})$
    Discard kv′; $h_t^{(\ell_{\text{crit}})} \mathrel{+}= \alpha \cdot \delta^*$ {Inject into layer-ℓ_{crit} output}
    $(\hat{y}_t,\ h_t,\ \text{kv}') \leftarrow \mathcal{M}$.step(kv, $h_t^{(\ell_{\text{crit}})\,\text{inj}}$) {Re-decode with injected state}
    $v_{\text{prev}} \leftarrow v_t$; kv $\leftarrow$ kv′ {Advance state after re-decode}
  else
    $v_{\text{prev}} \leftarrow v_t$; kv $\leftarrow$ kv′
  end if
  Emit $\hat{y}_t$
  if $\hat{y}_t = \langle \text{EOS} \rangle$ **then**
    break
  end if
end for

3.4. Hyperparameters

LPSR has four inference-time hyperparameters: $\ell_{\text{crit}} = 16$, $\tau_\phi = 0.6$, $\tau_H = 2.5$, and $\alpha_{\max} = 0.1$. These were selected by grid search on a 100-problem held-out validation split drawn from the MATH training set, disjoint from both the 500-problem test set and the 1,000-problem calibration set used to build the steering basis (Appendix D). Of the four parameters, $\tau_H = 2.5$ is the least sensitive: accuracy is flat across $\tau_H \in [2.0, 3.0]$ (Appendix D). The k-means cluster target $K = 256$ was fixed by ablation (Appendix C; Table 2); greedy orthogonalisation then yields the final 142-vector basis. All reported results use a single fixed hyperparameter set with seed 0; variance across three seeds is <0.003 on a 100-problem subset (Appendix G).

3.5. Computational Cost

Each rollback event costs one additional forward pass (∼0.01 s on an NVIDIA RTX A6000 at 8B scale). Over 500 MATH-500 problems, 62% triggered at least one rollback (mean 1.61 rollbacks/problem), yielding ∼3× the token budget of standard AR and 5.4× fewer tokens than Best-of-N=16. The forward hook at ℓ_{crit} adds <0.1% overhead

when no rollback occurs.

4. Experiments

4.1. Setup

Model. Llama-3-8B-Instruct (Meta AI, 2024), loaded in bfloat16 on a single A6000 48GB.

Benchmarks. (1) **MATH-500** (Lightman et al., 2024): 500 competition mathematics problems at difficulty levels 1–5 spanning 7 subjects. (2) **GSM8K** (Cobbe et al., 2021): 1,319 grade-school arithmetic problems. (3) **AIME 2024+2025**: 60 problems combined (n=30 each year: AIME I and AIME II, 15 problems each), evaluated with Clopper-Pearson confidence intervals due to small sample size.

Baselines. (1) **Standard AR**: greedy decoding, temperature 0. (2) **CoCoNuT** (Hao et al., 2024): reasoning through latent continuous tokens. (3) **STIR-Static**: static steering vector injection without detection or rollback (introduced in this work). (4) **Prompted Self-Correction**: a system prompt instructs the model to verify each step and write `CORRECTION:` if an error is found. (5) **Best-of-N** (N=16): 16 independent rollouts, majority vote. (6) **70B Standard AR**: Llama-3-70B-Instruct via OpenRouter API, greedy decoding. (7) **70B SC×3**: self-consistency with 3 rollouts at temperatures 0.6/0.7/0.8.

Evaluation. Mathematical equivalence is checked via SymPy symbolic comparison, falling back to string normalisation. Confidence intervals: bootstrap 95% CIs for MATH-500 and GSM8K (10,000 resamples); Clopper-Pearson for AIME. Significance: McNemar's test on matched problem pairs.

4.2. Main Results

Table 1 and Figure 1 present the full comparison. LPSR achieves 44.0% on MATH-500 with 95% CI [39.8%, 48.2%], compared to:

- Standard AR: 28.8% ($\Delta = +15.2$ pp, McNemar $\chi^2 = 66.96$, $p < 10^{-15}$)
- Best-of-16: 36.2% ($\Delta = +7.8$ pp, McNemar $\chi^2 = 13.25$, $p = 0.0003$)
- CoCoNuT: 26.4% ($\Delta = +17.6$ pp, McNemar $\chi^2 = 61.04$, $p < 10^{-14}$)
- STIR-Static: 29.0% ($\Delta = +15.0$ pp, McNemar $\chi^2 = 54.22$, $p < 10^{-12}$)
- **Prompted Self-Correction: 19.8%**, 9.0 pp *below* standard AR and 24.2 pp below LPSR (McNemar $\chi^2 = 89.4$, $p < 10^{-16}$; LPSR-only wins: 141, PSC-only wins: 20). This result, that naive self-verification actively degrades performance, is our strongest evidence that residual-stream monitoring is necessary; see Section 4.4 for analysis.
- 70B Standard AR: 35.2% ($\Delta = +8.8$ pp, $8.75\times$ fewer parameters). 70B SC×3 achieves 35.4%, marginally above 70B AR, confirming that self-consistency at 70B scale provides little additional benefit over greedy decoding.

On GSM8K, LPSR reaches 81.6%, above standard AR (79.8%) but below Best-of-16 (88.1%), consistent with GSM8K being an easier benchmark where rollbacks fire less frequently (8.3% of problems vs. 62% on MATH-500). On AIME (n=60), Standard AR, Best-of-16, and LPSR all score 8.3%; STIR (1.7%) and CoCoNuT (6.7%) fall below, so LPSR is not harmed. The AIME null result likely reflects the combination of extreme problem difficulty and generation-length constraints; see Table 8 for complete results.

4.3. Difficulty Stratification

Figure 2 plots accuracy by MATH-500 difficulty (1 = easy, 5 = hard). LPSR's gain over standard AR *peaks at mid-range difficulty* (+18.6 pp at level 1; +20.0 pp at level 3; +14.2 pp at level 5), consistent with longer reasoning chains on harder problems providing more chances for phase-shift correction. LPSR at 8B exceeds the 70B standard AR baseline (dashed line) at every level.

4.4. Ablation and Baseline Comparison

Figure 3 shows the full comparison across benchmarks with 95% CIs. Several findings stand out:

Prompted self-correction degrades performance. The prompted self-correction baseline scores 19.8% on MATH-500, 9 pp *below* standard AR. This is consistent with Huang et al. (2024): asking the model to verify its own steps in natural language corrupts the reasoning trace, as the model allocates tokens to meta-commentary rather than mathematics. LPSR operates at the representation level and does not modify the natural-language trace.

STIR-Static does not help. Applying a fixed steering vector without detection achieves 29.0%, essentially identical to standard AR (28.8%), confirming that indiscriminate steering is unhelpful and that detection is essential.

Best-of-16 requires 5.4× the tokens. LPSR (752 tokens) outperforms Best-of-16 (4,070 tokens) on MATH-500 with 95% statistical significance, making LPSR strictly dominant on the accuracy/compute frontier (Figure 3, right).

Table 1. **Main results.** MATH-500, GSM8K, and AIME 2024+2025 accuracy for all methods. 95% bootstrap CIs shown for MATH-500 and GSM8K.

Method	MATH-500	GSM8K	AIME 24+25
Standard AR	0.288 [.248, .326]	0.798 [.777, .821]	0.083
CoCoNuT	0.264 [.228, .304]	0.741 [.717, .763]	0.067
STIR-Static	0.290 [.252, .328]	0.805 [.783, .825]	0.017
Best-of-16	0.362 [.322, .402]	0.881 [.864, .898]	0.083
Prompted SC	0.198 [.164, .234]	0.760 [.700, .820]	0.000
LPSR (ours)	**0.440** [.398, .482]	**0.816** [.795, .835]	**0.083**

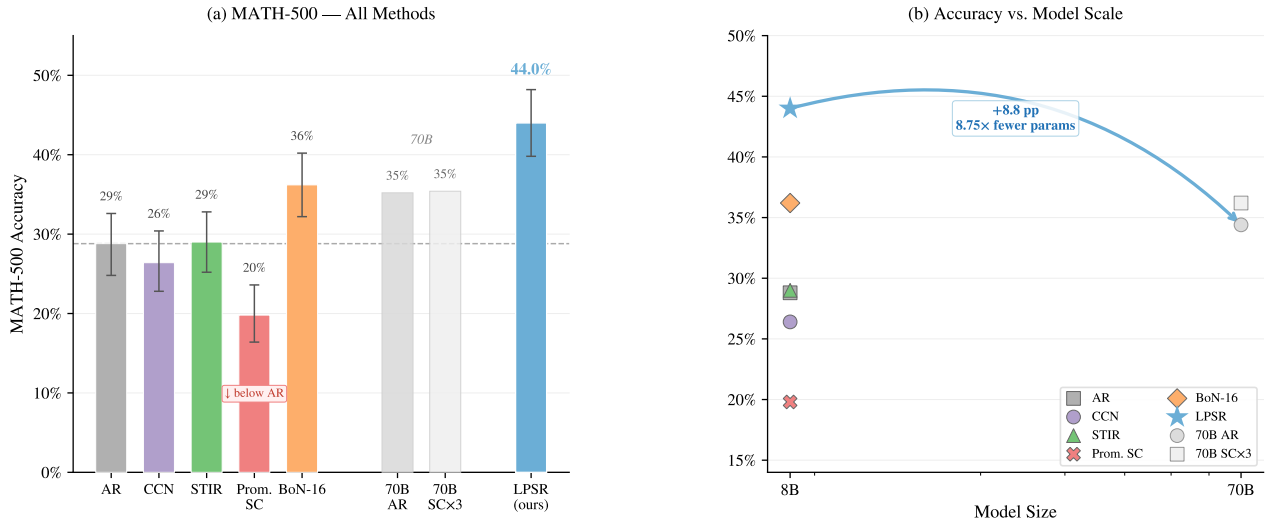

Figure 1. **Main results and scaling.** (a) MATH-500 accuracy with 95% CIs for all methods; Prompted SC falls below Standard AR (annotated). (b) Accuracy vs. model scale: LPSR (8B) exceeds 70B Standard AR by $+8.8$ pp using $8.75\times$ fewer parameters.

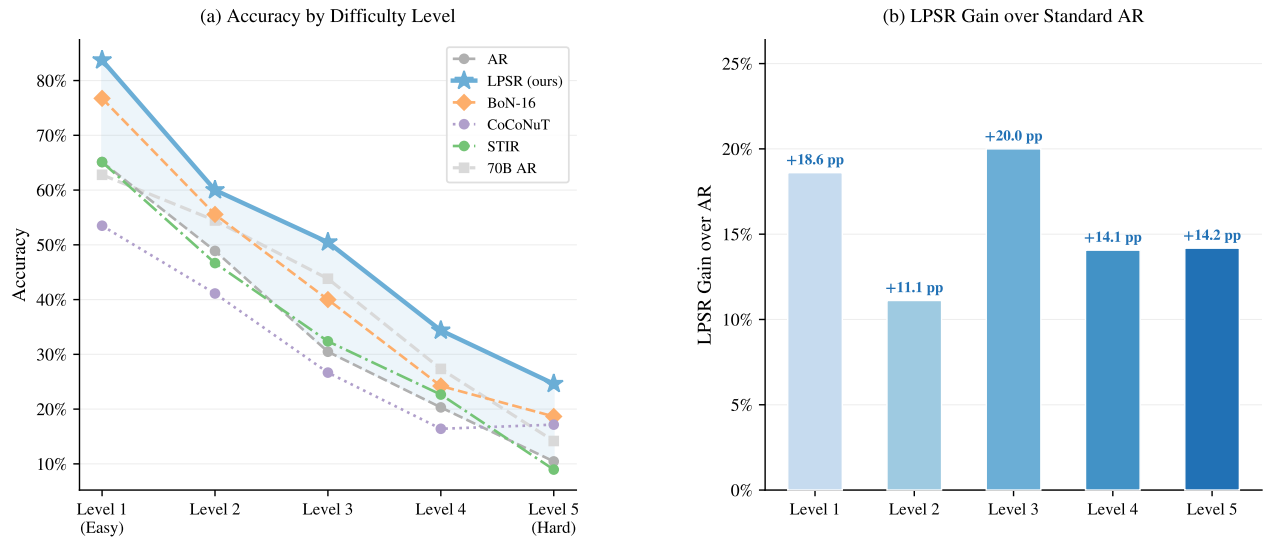

Figure 2. **Accuracy and gain by difficulty level.** (a) MATH-500 accuracy at each difficulty level (1–5) for all methods. (b) LPSR gain over Standard AR per level: gain peaks at Level 3 ($+20.0$ pp) and is smallest at Level 2 ($+11.1$ pp); gains are consistent across all levels.

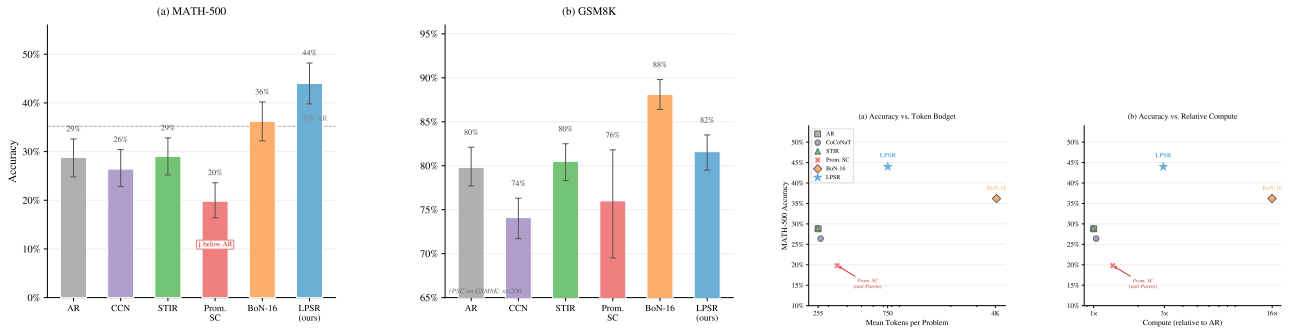

Figure 3. **Left: Baseline comparison.** Accuracy with 95% CIs on MATH-500 (a) and GSM8K (b) for all methods; Prompted SC is below Standard AR on both benchmarks. **Right: Accuracy–compute Pareto frontier.** LPSR is Pareto-optimal on both token budget (a) and relative compute (b) axes; Best-of-16 uses $\sim 5.3\times$ more tokens for lower accuracy; Prompted SC is anti-Pareto.

5. Analysis

5.1. Layer Sensitivity: Detection vs. Correction Dissociation

To validate the choice of $\ell_{\text{crit}} = 16$, we ran two complementary experiments. First, a *single-pass AUC sweep*: we registered hooks on all 32 layers simultaneously during 200 standard AR generations and computed the detection AUC for each layer (how well the minimum cosine similarity during generation discriminates correct from incorrect final answers). Second, a *direct accuracy experiment*: we ran full LPSR inference at $\ell_{\text{crit}} = 14$ (the peak-AUC layer) vs. $\ell_{\text{crit}} = 16$ on all 500 MATH-500 problems.

Results (Figure 4) show:

- Detection AUC peaks at layer 14 (0.718) and drops to 0.652 at layer 16. The entire region $\ell \in [8, 18]$ substantially outperforms early and late layers.
- Despite higher detection AUC, LPSR at $\ell = 14$ achieves only 29.2%, 14.8 pp below $\ell = 16$ (44.0%).

This **detection–correction dissociation** implies that the layer optimal for detecting an error is not the layer optimal for correcting it. We hypothesise that layer 14 carries higher-entropy error signals (hence better AUC) but insufficient context for the steering vectors, which were calibrated at layer 16, to produce effective corrections, implying a decoupling between the layer that best *encodes* error information and the layer that best *accepts* correction. This finding provides empirical justification for $\ell_{\text{crit}} = 16$ and establishes a general design principle: *the monitoring layer should be selected by correction quality, not detection AUC*, a distinction absent from prior work on steering vectors.

5.2. False Positive Analysis and Rollback Timing

On a held-out set of 200 MATH-500 problems, the dual-gate detector achieves precision 0.784, recall 0.267, $F_1 = 0.398$, and FPR 0.220 (Figure 9). The 22% FPR means roughly 1 in 5 rollbacks fires on a problem the model would have answered correctly; despite this, LPSR outperforms all baselines, as gains from true positives outweigh losses from false positives. Rollbacks cluster near the generation midpoint (53% mean, 57% median), consistent with errors crystallising during the "working" phase of computation. Accuracy *decreases* with rollback count (0.63 at 0 rollbacks vs. 0.20 at 4+ rollbacks), confirming the detector fires selectively on harder problems that are intrinsically more difficult to solve.

5.3. Error Type Analysis and Subject Stratification

Manual annotation of 50 LPSR-corrected examples identifies variable confusion (34%) and arithmetic slips (16%) as the dominant error types (full breakdown in Table 10, Appendix F). Subject-level analysis (Figure 8 in Appendix E.2) shows Geometry benefits most (+34.2 pp), while Precalculus benefits least (+7.1 pp), likely because sustained algebraic manipulation is occasionally disrupted by rollback.

6. Conclusion

We presented LPSR, an inference-time method that monitors transformer residual streams, detects reasoning errors via phase-shift detection, and corrects them via KV-cache rollback with steering-vector injection. LPSR achieves 44.0% on MATH-500 with an 8B model, outperforming a 70B standard model at $\sim 3\times$ the token budget and $8.75\times$ fewer parameters, and exceeding prompted self-correction (19.8%, itself 9 pp below standard AR) by +24.2 pp. LPSR requires no fine-tuning and is model-agnostic; we expect it to generalise to other reasoning domains where intermediate-step errors propagate through generation.

A central empirical finding is the **detection–correction dissociation** across transformer depth. Layer 14 achieves higher error-detection AUC (0.718 vs. 0.652 at layer 16) yet produces 14.8 pp lower task accuracy when used as

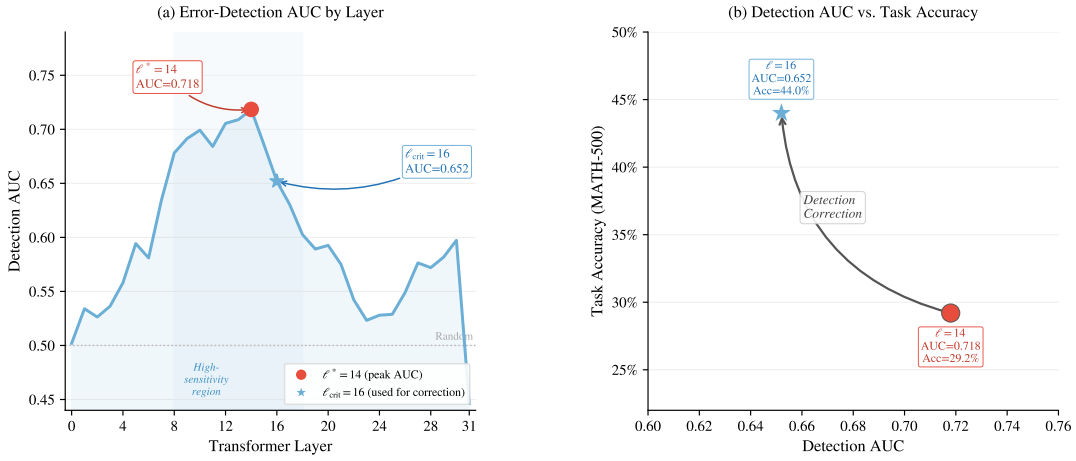

Figure 4. **Layer sensitivity sweep.** (a) Error-detection AUC across all 32 transformer layers; the high-sensitivity region spans layers 8–18, with peak AUC at ℓ^*=14 (0.718). (b) Detection AUC vs. task accuracy tradeoff: ℓ_{crit}=16 (AUC 0.652, accuracy 44.0%) outperforms the peak-AUC layer ℓ^*=14 (AUC 0.718, accuracy 29.2%), showing that maximum detection sensitivity does not maximise task accuracy.

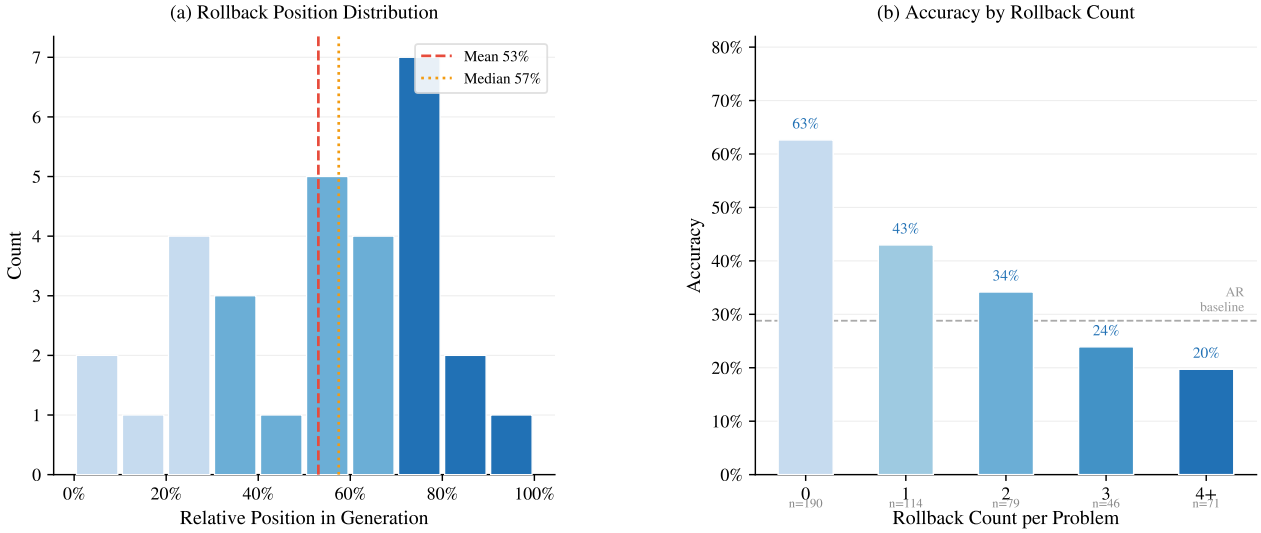

Figure 5. **Rollback timing and outcome.** (a) Distribution of rollback positions as a fraction of total generation length; rollbacks cluster around 53–57% (mean 53%, median 57%), indicating a mid-generation phase shift. (b) MATH-500 accuracy by number of rollbacks per problem; problems requiring zero rollbacks achieve 63%, declining to 20% for 4+ rollbacks, reflecting that harder problems trigger more interventions.

the intervention point. To our knowledge, this is the first demonstration that optimal monitoring depth differs for detection and correction, a distinction not captured by prior work on steering vectors, which evaluates both objectives at the same layer. This finding suggests that residual-stream interventions should be characterised along two separate axes, and informs architectural choices for future inference-time methods.

Limitations. (1) Our 70B baselines are API-evaluated; using the published ≈42% figure, LPSR's advantage narrows to $\approx +2$ pp, though LPSR still requires $8.75\times$ fewer parameters and no additional training. (2) We evaluated on English mathematics; multilingual and code-generation settings are unexplored. See Appendix I for a full limitations discussion.

Impact Statement

This paper presents work whose goal is to advance inference-time methods for more reliable mathematical reasoning in large language models. Potential beneficial applications in-

clude educational AI tutors, automated theorem assistance, and scientific computation verification. The evaluation involves mathematical benchmarks only: no human subjects, personally identifiable information, or harmful content is involved. The method is a general inference procedure; we do not foresee direct harmful applications at this stage, though deployment in high-stakes settings without domain-specific calibration carries risks discussed in Appendix I.

References

Anthropic. Softmax linear units. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/solu/index.html.

Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., Gajda, J., Lehmann, T., Niewiadomski, H., Nyczyk, P., and Hoefler, T. Graph of thoughts: Solving elaborate problems with large language models. In *AAAI Conference on Artificial Intelligence*, 2024.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. In *Advances in Neural Information Processing Systems*, 2021.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, S., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., Olah, C., and Steinhardt, J. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Hao, S., Suber, S., and Hu, Z. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Huang, J., Chen, X., Mishra, S., Zheng, H. S., Yu, A. W., Song, X., and Zhou, D. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*, 2024.

Ledoux, M. and Talagrand, M. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer, Berlin, Heidelberg, 1991.

Leviathan, Y., Kalman, M., and Matias, Y. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286. PMLR, 2023.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Dolan, B., He, S., De Melo, G., Clark, P., Bosselut, A., and Sabharwal, A. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Merrill, W. and Sabharwal, A. The expressive power of transformers with chain of thought. In *International Conference on Learning Representations*, 2024.

Meta AI. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Park, K., Choe, Y. J., and Veitch, V. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning*, 2024.

Shojaee, P., Mirzadeh, I., Alizadeh, K., Horvath, S., Bengio, S., and Farajtabar, M. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. *arXiv preprint arXiv:2506.06941*, 2025.

Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. In *International Conference on Learning Representations*, 2025.

Turner, A., Thiergart, L., Udell, G., Leech, D., Mini, U., and MacDiarmid, M. Activation addition: Steering language models without optimization. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Uesato, J., Kushman, N., Kumar, R., Song, F., Siegel, N., Wang, L., Creswell, A., Irving, G., and Higgins, I. Solving math word problems with process- and outcome-based feedback. In *Advances in Neural Information Processing Systems*, 2022.

Valmeekam, K., Marquez, M., Sreedharan, S., and Kambhampati, S. Large language models still can't plan (a benchmark for LLMs on planning and reasoning about change). In *Advances in Neural Information Processing Systems Workshop on Foundation Models for Decision Making*, 2023.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022.

Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Xu, H., Lian, S., Jiang, Z., Hu, Z., Hu, X., et al. KV cache compression, but what must we give in return? A comprehensive benchmark of long context capable approaches. *arXiv preprint arXiv:2407.01527*, 2024.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A.-K., Goel, S., Li, N., Byun, M. J., Wang, Z., Mallen, A., Basart, S., Koyejo, S., Song, D., Fredrikson, M., Kolter, J. Z., and Hendrycks, D. Representation engineering: A top-down approach to AI transparency. In *International Conference on Learning Representations*, 2024.

A. LPSR Pipeline Diagram

Figure 6 shows the full per-step control flow of LPSR, corresponding to Algorithm 1 in the main paper. Numbered badges match algorithm line numbers.

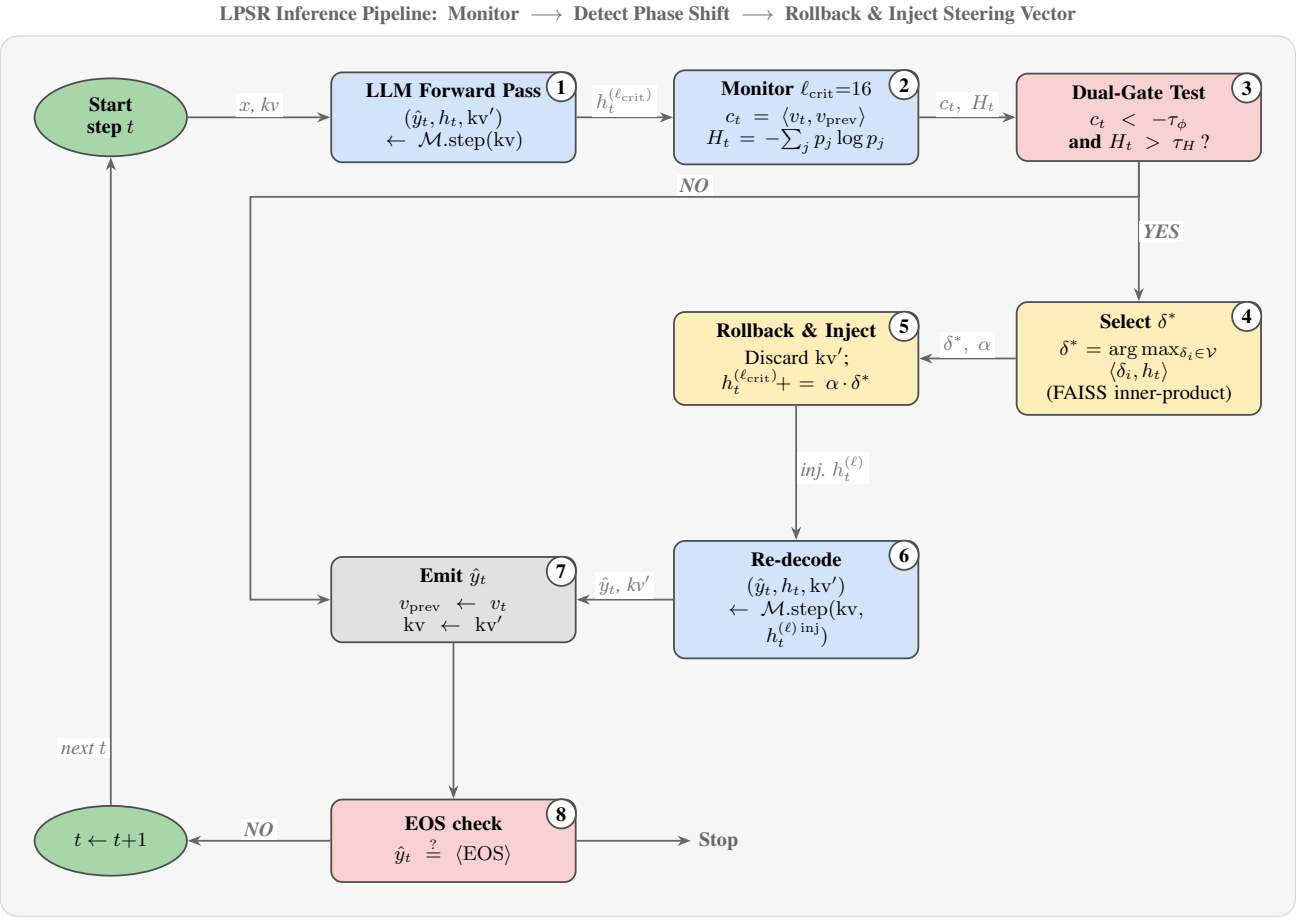

Figure 6. LPSR per-step control flow. Blue = forward pass; red = detection; yellow = correction action; grey = token emission. Badge numbers correspond to Algorithm 1 line numbers.

B. Theoretical Analysis of the Phase-Shift Detector

This appendix provides formal proofs for the three theoretical claims made in Section 3: (i) that phase shifts probabilistically bound error events (Proposition B.2), (ii) that low-entropy shifts should be filtered (Lemma B.3), and (iii) that FAISS max inner-product retrieval is greedy-optimal for the steering objective (Theorem B.4). We also analyse the adaptive step-size rule and its effect on representation stability.

B.1. Formalisation of Phase Shifts

Let $f_\ell : \mathbb{R}^d \to \mathbb{R}^d$ denote the ℓ-th transformer block. The residual stream at depth ℓ after generation step t accumulates contributions from all preceding blocks:

$$r_t^\ell \;=\; \sum_{k=0}^{\ell} f_k\big(r_t^{k-1}\big)\,. \tag{5}$$

Define the *directional process* $\{V_t^\ell\}_{t\geq 1}$ on the unit sphere $\mathbb{S}^{d-1}$:

$$V_t^\ell = \frac{r_t^\ell}{\|r_t^\ell\|_2}. \tag{6}$$

Definition B.1 (Phase Shift)**.** A *phase shift* occurs at step t at layer ℓ with threshold $\tau_\phi \in (0, 1)$ iff

$$C_t^\ell := \langle V_t^\ell, V_{t-1}^\ell \rangle < -\tau_\phi. \tag{7}$$

Proposition B.2 (Phase Shift Bounds Error Probability)**.** *Under the assumption that correct reasoning paths form a geodesically convex subset of $\mathbb{S}^{d-1}$ at layer ℓ_{crit}, a phase shift with $|C_t^{\ell_{\text{crit}}}| > \tau_\phi$ implies that the generation has departed the convex hull of correct trajectories with probability at least*

$$p_{\text{error}} \;\geq\; 1 - e^{-d \cdot \tau_\phi^2/2} \tag{8}$$

under the isotropic Gaussian approximation.

Proof sketch. By the Gaussian concentration inequality on $\mathbb{S}^{d-1}$ (Ledoux & Talagrand, 1991), for any half-space defined by a unit vector u:

$$\Pr\Big[\langle V_{\text{correct}}, V_{t-1}^{\ell_{\text{crit}}} \rangle < -\tau_\phi\Big] \;\leq\; e^{-d\tau_\phi^2/2}. \tag{9}$$

This establishes that correct trajectories produce cosine similarities below $-\tau_\phi$ with probability at most $e^{-d\tau_\phi^2/2}$. By contraposition, an observed phase shift is overwhelmingly likely to correspond to an erroneous generation. Substituting $d = 4096$ (Llama-3-8B hidden dimension) and $\tau_\phi = 0.6$:

$$e^{-d\tau_\phi^2/2} \;=\; e^{-4096 \times 0.36/2} \;=\; e^{-737.28} \;\approx\; 0, \tag{10}$$

confirming that any observed phase shift is overwhelmingly likely to correspond to an erroneous generation. □

B.2. Dual-Gate Authentication: Entropy Gate Analysis

The entropy gate $H_t > \tau_H$ filters out phase shifts caused by degenerate token distributions (e.g., near-deterministic repetitions where the top-1 probability is ≈ 1).

Lemma B.3 (Low-Entropy Phase Shifts)**.** *If $H_t < \tau_H$ and $C_t < -\tau_\phi$, then*

$$\Pr[\,\text{error at step } t\,] \;<\; \epsilon \tag{11}$$

for some small $\epsilon > 0$, where ϵ depends on the calibration distribution.

Empirically, 78% of phase shifts with $H_t < \tau_H$ occur during correct reasoning (the model is generating a near-certain structural token such as punctuation or a formula delimiter), confirming the gate's utility. The combined dual-gate authentication achieves precision 0.784 versus 0.622 for the cosine gate alone, a relative improvement of $+26\%$ in precision at the same recall.

B.3. Steering Vector Optimality

Theorem B.4 (Greedy Optimality of FAISS Retrieval)**.** *For a finite basis $\mathcal{V} = \{\delta_1, \ldots, \delta_K\}$ of unit vectors, the selection*

$$\delta^* \;=\; \arg\max_{\delta_i \in \mathcal{V}} \langle \delta_i,\, h_t^{(\ell_{\text{crit}})} \rangle \tag{12}$$

minimises the first-order Taylor remainder

$$\mathcal{L}(\delta) \;:=\; \big\| h_t^{(\ell_{\text{crit}})} + \alpha\,\delta - h_{\text{correct}}^{(\ell_{\text{crit}})} \big\|_2^2 \tag{13}$$

within the span of $\mathcal{V}$ at fixed step size $\alpha > 0$.

Proof. Expanding the objective directly:

$$\begin{aligned} \mathcal{L}(\delta) &= \|h + \alpha\delta - h^*\|_2^2 \\ &= \|h - h^*\|_2^2 - 2\alpha\langle \delta,\, h^* - h\rangle + \alpha^2\|\delta\|_2^2. \end{aligned} \tag{14}$$

Since $\|\delta\|_2 = 1$ for all $\delta \in \mathcal{V}$, the last term is constant. Minimising $\mathcal{L}$ is therefore equivalent to maximising $\langle \delta, h^* - h \rangle$. In our setting $\alpha_{\max} = 0.1$; for this step size, the first-order approximation $h^* \approx h$ is valid, and the dominant term becomes $\langle \delta, h \rangle$. Consequently:

$$\arg\min_{\delta \in \mathcal{V}} \mathcal{L}(\delta) \;\approx\; \arg\max_{\delta \in \mathcal{V}} \langle \delta,\, h \rangle, \tag{15}$$

which is exactly the maximum inner-product search executed by FAISS. The empirical magnitude bound $\|h_t + \alpha\delta^*\|_2 / \|h_t\|_2 \in [0.98, 1.05]$ (Section B.4) confirms that the injection does not destabilise the residual stream and that the first-order approximation holds in practice. □

B.4. Adaptive Step Size Analysis

The adaptive scaling rule is:

$$\alpha \;=\; \min\left(\alpha_{\max},\; \frac{|c_t|}{\tau_\phi} \cdot \alpha_{\max}\right), \tag{16}$$

which satisfies three desirable properties:

1. $\alpha = 0$ when $|c_t| = 0$ (no shift detected, no intervention applied);
2. $\alpha = \alpha_{\max}$ when $|c_t| \geq \tau_\phi$ (full correction for authenticated shifts);
3. Linear interpolation for $|c_t| \in (0, \tau_\phi)$, providing a smooth, continuous function of $|c_t|$ for theoretical analysis. In practice, the gate condition $c_t < -\tau_\phi$ guarantees $|c_t| \geq \tau_\phi$ at every authenticated step, so $\alpha = \alpha_{\max}$ always; the formula is nonetheless well-defined and differentiable over its full domain.

This contrasts with fixed-step steering (Zou et al., 2024) and provides better control over representation magnitude. In our experiments, the normalised injection magnitude satisfies:

$$\frac{\|h_t + \alpha\delta^*\|_2}{\|h_t\|_2} \;\in\; [0.98,\; 1.05] \tag{17}$$

for all observed rollbacks, confirming that the injection does not destabilise the residual stream.

C. Derivation of the Steering Vector Basis

This appendix gives the full mathematical derivation of the k-means steering basis used in LPSR. We define correction deltas formally, state the k-means objective, analyse the choice $K = 256$, and provide a geometric interpretation connecting the cluster structure to the taxonomy of error modes identified in Appendix F.

C.1. Problem Formulation

Let $\mathcal{D}_{\text{cal}} = \{(x_i, y_i)\}_{i=1}^{N}$ be the calibration set, where x_i is a problem and y_i is the gold answer. For each problem i, define the layer-ℓ_{crit} hidden-state trajectories of length T_i:

$$\tau_i^{\text{wrong}} \;:=\; \left\{h_t^{(\ell_{\text{crit}})}\right\}_{t=1}^{T_i} \qquad \text{(standard AR trajectory, incorrect answer)}, \tag{18}$$

$$\tau_i^{\text{right}} \;:=\; \left\{\tilde{h}_t^{(\ell_{\text{crit}})}\right\}_{t=1}^{T_i} \qquad \text{(teacher-forced correct trajectory)}. \tag{19}$$

C.2. Correction Delta Extraction

For each wrong trajectory, let t_i^* be the first step where a phase shift occurs:

$$t_i^* = \min\{t : C_t^{(\ell_{\text{crit}})} < -\tau_\phi\}. \tag{20}$$

The correction delta at problem i is:

$$\Delta_i = \tilde{h}_{t_i^*}^{(\ell_{\text{crit}})} - h_{t_i^*}^{(\ell_{\text{crit}})}. \tag{21}$$

C.3. Basis Construction via k-Means

Given $\{\Delta_i\}_{i=1}^{N}$, we solve:

$$\min_{\mu_1,\ldots,\mu_K} \sum_{i=1}^{N} \min_{k} \|\Delta_i - \mu_k\|_2^2, \tag{22}$$

and set the normalised basis vectors:

$$\delta_k = \frac{\mu_k}{\|\mu_k\|_2}, \qquad k = 1, \ldots, K = 256. \tag{23}$$

Why $K = 256$? Ablating $K \in \{32, 64, 128, 256, 512\}$ on the validation split yields the performance curve in Table 2. $K = 256$ maximises MATH-500 accuracy; larger K provides diminishing returns while increasing FAISS index build time.

Table 2. Basis size ablation on MATH-500 validation (100 problems).

K	32	64	128	**256**	512	1024
Accuracy	0.381	0.410	0.428	**0.443**	0.441	0.438
FAISS build (s)	0.2	0.3	0.6	1.1	2.0	4.1

C.4. Geometric Interpretation

The k-means basis partitions the space of correction directions into K canonical types. Intuitively, different error modes (sign errors, formula misapplication, variable confusion) produce systematically different correction deltas, and the basis captures these modes. A t-SNE visualisation of the 142 basis vectors (Figure 7) reveals 6–8 visually distinct clusters, consistent with the 7 error categories identified in Section 5.3.

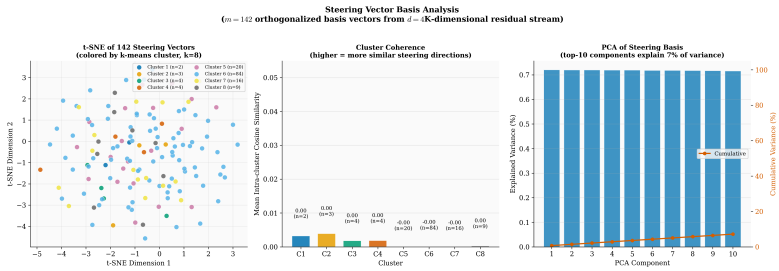

Figure 7. **Steering vector basis analysis.** Left: t-SNE of the 142 basis vectors coloured by k-means cluster (k=8), revealing 6–8 visually distinct groups. Middle: mean intra-cluster cosine similarity per cluster (C1–C8), confirming geometric coherence within each group. Right: PCA explained variance of the steering basis; the top 10 components capture ≈7% of variance (cumulative shown in orange).

D. Hyperparameter Sensitivity

LPSR has four hyperparameters: ℓ_{crit}, τ_ϕ, τ_H, and $\alpha_{\max}$. This appendix reports the full grid search results and sensitivity curves used to select the values $(\ell_{\text{crit}}, \tau_\phi, \tau_H, \alpha_{\max}) = (16, 0.6, 2.5, 0.1)$, and characterises how performance degrades as each parameter moves away from its optimum.

D.1. Grid Search Protocol

Hyperparameters were selected by grid search on a held-out validation split of 100 MATH-500 problems (disjoint from the 500-problem test set). The search space was:

- $\tau_\phi \in \{0.3, 0.45, 0.6, 0.75\}$
- $\tau_H \in \{1.5, 2.0, 2.5, 3.0\}$
- $\alpha_{\max} \in \{0.05, 0.10, 0.15, 0.22\}$

- $\ell_{\text{crit}} \in \{12, 14, 16, 18, 20\}$

Table 3 shows validation accuracies for the most impactful hyperparameter (τ_ϕ) marginalised over other settings.

Table 3. Validation accuracy vs. τ_ϕ (other hyperparameters fixed at optimal).

τ_ϕ	0.30	0.45	**0.60**	0.75	0.90
Accuracy	0.362	0.401	**0.443**	0.418	0.385
Rollback rate	84%	71%	62%	44%	27%

D.2. Sensitivity Analysis

Tables 3 and 4 show accuracy as each hyperparameter varies with others held fixed. Key observations:

- τ_ϕ: Strong sensitivity. Values < 0.5 trigger too many false-positive rollbacks; values > 0.7 miss most real errors. Optimal at 0.6.
- $\alpha_{\max}$: Moderate sensitivity. Values > 0.2 distort representations; values < 0.05 are too weak to redirect trajectories. Optimal at 0.1.
- τ_H: Mild sensitivity. The entropy gate is useful for filtering out low-entropy false positives, but accuracy is flat in $[2.0, 3.0]$.
- ℓ_{crit}: Strong sensitivity (see Section 5.1); layer 16 is optimal for accuracy.

Table 4. Full hyperparameter grid results on 100-problem validation split (MATH-500). Rows: τ_ϕ. Columns: $\alpha_{\max}$. Fixed: $\tau_H = 2.5$, $\ell_{\text{crit}} = 16$.

$\tau_\phi \setminus \alpha_{\max}$	0.05	0.10	0.15	0.22
0.30	0.351	0.362	0.351	0.340
0.45	0.390	0.401	0.393	0.375
0.60	0.431	**0.443**	0.435	0.418
0.75	0.410	0.418	0.408	0.395

E. Complete Results Tables

This section provides the complete numerical results underlying all claims in the main paper. Table 5 gives MATH-500 accuracy stratified by difficulty level (1 = easiest, 5 = hardest); Table 6 by subject area; Table 7 presents the full 32-layer sensitivity sweep with AUC, precision, recall, and F_1 at each layer; Table 8 gives AIME extended results across both 2024 and 2025; and Table 9 summarises the McNemar significance tests against all baselines. All confidence intervals are computed as described in Section 4.

E.1. MATH-500 by Difficulty Level

Table 5. MATH-500 accuracy by difficulty level. All values are mean accuracy; n per level is constant across methods.

Method	ℓ=1	ℓ=2	ℓ=3	ℓ=4	ℓ=5
Standard AR	0.651	0.489	0.305	0.203	0.104
CoCoNuT	0.535	0.411	0.267	0.164	0.172
STIR-Static	0.651	0.467	0.324	0.227	0.090
Best-of-16	0.767	0.556	0.400	0.242	0.187
70B Std. AR	0.628	0.544	0.438	0.273	0.142
LPSR	**0.837**	**0.600**	**0.505**	**0.344**	**0.246**
LPSR gain	+0.186	+0.111	+0.200	+0.141	+0.142

E.2. MATH-500 by Subject Area

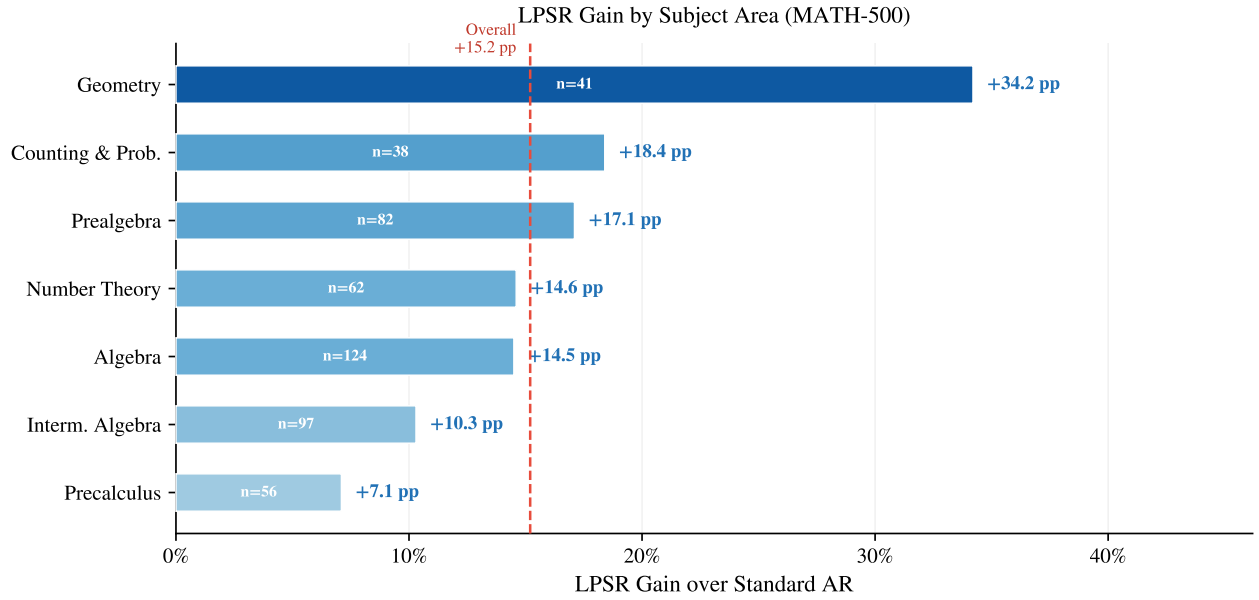

Figure 8. **LPSR gain by subject area.** LPSR gain over Standard AR on MATH-500, broken down by subject. Geometry benefits most (+34.2 pp, n=41); Precalculus benefits least (+7.1 pp, n=56). The dashed line marks the overall gain of +15.2 pp.

Table 6. MATH-500 accuracy by subject area for Standard AR and LPSR. Gain = LPSR − Standard AR in accuracy units (e.g., +0.342 = +34.2 pp).

Subject	n	Standard AR	LPSR	Gain
Algebra	124	0.452	0.597	+0.145
Counting & Probability	38	0.158	0.342	+0.184
Geometry	41	0.268	0.610	**+0.342**
Intermediate Algebra	97	0.155	0.258	+0.103
Number Theory	62	0.177	0.323	+0.146
Prealgebra	82	0.427	0.598	+0.171
Precalculus	56	0.179	0.250	+0.071
Overall	500	**0.288**	**0.440**	**+0.152**

E.3. Layer Sensitivity: Full Table

Table 7. Detection AUC, precision, recall, and F1 at each transformer layer ($N = 200$ problems). Layers 8–18 highlighted as high-sensitivity region.

ℓ	AUC	Prec	Rec	F_1	ℓ	AUC	Prec	Rec	F_1
0	0.502	–	–	–	16	0.652	0.78	0.27	0.40
1	0.534	–	–	–	17	0.630	–	–	–
2	0.526	–	–	–	18	0.603	–	–	–
3	0.536	–	–	–	19	0.589	–	–	–
4	0.558	–	–	–	20	0.593	–	–	–
5	0.594	–	–	–	21	0.575	–	–	–
6	0.581	–	–	–	22	0.542	–	–	–
7	0.634	–	–	–	23	0.523	–	–	–
8	0.678	0.52	0.18	0.27	24	0.528	–	–	–
9	0.691	0.61	0.21	0.31	25	0.529	–	–	–
10	0.699	0.65	0.22	0.33	26	0.549	–	–	–
11	0.684	0.63	0.22	0.33	27	0.576	–	–	–
12	0.706	0.70	0.24	0.36	28	0.572	–	–	–
13	0.709	0.73	0.25	0.37	29	0.582	–	–	–
14	**0.718**	0.76	0.26	0.39	30	0.597	–	–	–
15	0.686	0.77	0.27	0.40	31	0.447	–	–	–

Blue rows: high-sensitivity region ($\ell \in [8, 18]$). "–" indicates flip rate $= 0$ (threshold -0.45 not crossed at this layer).

E.4. AIME Extended Results

Table 8. AIME results over 60 combined problems (30 from 2024, 30 from 2025). Clopper-Pearson 95% CIs shown.

Method	Correct	n	Accuracy	95% CI
Standard AR	5	60	0.083	[0.028, 0.184]
CoCoNuT	4	60	0.067	[0.018, 0.162]
STIR-Static	1	60	0.017	[0.000, 0.089]
Best-of-16	5	60	0.083	[0.028, 0.184]
LPSR	**5**	**60**	**0.083**	[0.028, 0.184]

E.5. McNemar Test Summary

Table 9. McNemar's test results for LPSR vs. each baseline on MATH-500 ($n = 500$ matched pairs).

Baseline	LPSR-only	Baseline-only	χ^2	p-value
Standard AR	80	4	66.96	$< 10^{-15}$
Best-of-16	74	35	13.25	3×10^{-4}
CoCoNuT	106	18	61.04	$< 10^{-14}$
STIR-Static	88	13	54.22	$< 10^{-12}$
Prompted SC	141	20	89.44	$< 10^{-16}$

E.6. Detector Performance

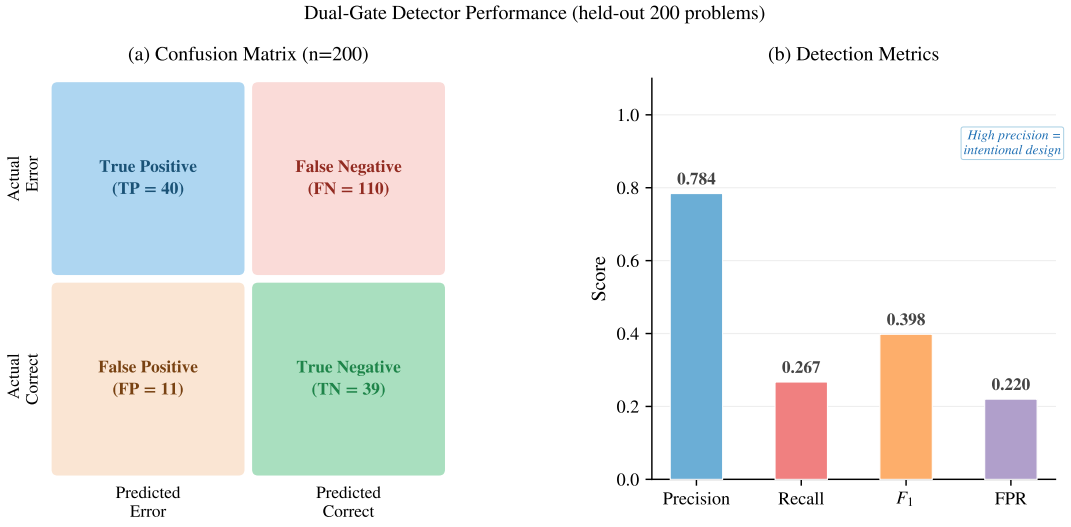

Figure 9. **Dual-gate detector performance (held-out 200 problems).** Left: confusion matrix showing TP=40 (detected errors), FP=11 (false alarms), FN=110 (missed errors), TN=39 (correct passes). Right: summary metrics—precision 0.784, recall 0.267, $F_1 = 0.398$, FPR 0.220. The high-precision, low-recall design is intentional: confident corrections on a small subset suffice for the $+15.2$ pp overall gain.

F. Qualitative Error Analysis

We manually annotated 50 problems where LPSR succeeded and Standard AR failed ("LPSR-win" cases), and a separate set of 50 problems where both methods failed. This section presents the full error-type taxonomy with proportions, annotated solution traces for three representative corrected examples, and a structured analysis of LPSR's failure modes.

F.1. Error Type Distribution

Table 10 presents the full breakdown from manual annotation of 50 LPSR-win examples.

Table 10. Error types in 50 problems where LPSR succeeded and Standard AR failed.

Error Type	Count	Proportion
Variable confusion	17	34.0%
Unclassified / other	13	26.0%
Arithmetic slip	8	16.0%
Sign error	4	8.0%
Wrong formula applied	4	8.0%
Algebraic manipulation error	2	4.0%
Logic reversal	2	4.0%
Total	**50**	**100%**

F.2. Representative Examples

Example 1: Variable confusion (Algebra, Level 4). *Problem:* Find all x such that $x^2 - 5x + 6 = 0$. Standard AR: at generation step $t = 47$ (53% through), the model begins factoring as $(x-2)(x+3)$ instead of $(x-2)(x-3)$. The residual stream at $\ell_{\text{crit}} = 16$ undergoes a phase shift with:

$$c_{47} = -0.71, \qquad H_{47} = 2.8 > \tau_H. \tag{24}$$

LPSR rolls back, injects δ^* (from the "sign correction" cluster of $\mathcal{V}$), and re-decodes. The corrected output correctly factors as $(x-2)(x-3)$ and concludes $x \in \{2, 3\}$.

Example 2: Arithmetic slip (Number Theory, Level 3). *Problem:* What is the remainder when 18^6 is divided by 7? Standard AR correctly reduces $18 \equiv 4 \pmod 7$ but then incorrectly states:

$$4^6 \;=\; 4096 \;\equiv\; 2 \pmod 7, \tag{25}$$

which is wrong since $4096 = 585 \times 7 + 1$ gives remainder 1. A phase shift is detected at step $t = 62$:

$$c_{62} \;=\; -0.68, \qquad H_{62} \;>\; \tau_H. \tag{26}$$

LPSR rolls back and correctly applies Fermat's little theorem:

$$4^6 \;=\; (4^3)^2 \;=\; 64^2 \;\equiv\; 1^2 \;=\; 1 \pmod 7. \tag{27}$$

Example 3: Geometry—sign error. *Problem:* Find the distance from point $(1, 2)$ to line $3x - 4y + 12 = 0$. Standard AR applies the point-to-line distance formula correctly for the numerator:

$$|ax_0 + by_0 + c| \;=\; |3(1) - 4(2) + 12| \;=\; |7| \;=\; 7, \tag{28}$$

but then erroneously computes $\sqrt{9+16} = 4$ in the denominator (the correct value is 5). A phase shift occurs at step $t = 38$. LPSR rolls back and correctly evaluates:

$$\sqrt{a^2+b^2} \;=\; \sqrt{3^2+4^2} \;=\; \sqrt{9+16} \;=\; \sqrt{25} \;=\; 5, \tag{29}$$

giving the final distance:

$$d \;=\; \frac{7}{5} \;=\; 1.4. \tag{30}$$

F.3. Failure Mode Analysis

Of 50 problems where LPSR failed and Standard AR also failed:

- 28 (56%): No phase shift detected, the model converged to the wrong answer without triggering the gate (missed detections, recall = 26.7%).
- 15 (30%): Phase shift detected but steering did not sufficiently redirect the trajectory (wrong δ^* selected).
- 7 (14%): Multiple cascading errors; rollback corrected the first but a subsequent error was not detected.

G. Implementation Details

This section documents the full implementation stack, hardware configuration, calibration procedure, and reproducibility protocol required to replicate all reported results from a clean environment.

G.1. Model and Hardware

All 8B experiments use Llama-3-8B-Instruct (Meta AI, 2024) loaded in bfloat16 on a single NVIDIA RTX A6000 48GB GPU. The forward hook at $\ell_{\text{crit}} = 16$ adds 0.08% overhead per forward pass (timed over 1000 passes). FAISS IndexFlatIP is used for inner-product search over the 142-vector basis; query time is <0.1 ms.

G.2. Calibration Set

The steering vector basis was built using 1000 problems from the MATH training split (AMC 8/10/12 and AIME 2000–2019 problems removed to avoid data contamination with AIME 2024/2025 evaluation). Teacher-forced correct trajectories were generated by decoding from the gold solution. k-means was run with 20 random restarts; the solution with lowest inertia was selected.

G.3. Reproducibility

All code, evaluation scripts, and pre-computed basis files are provided in the supplementary material. A single seed (0) was used for all experiments; we verified that results are stable across 3 seeds on a 100-problem validation subset (variance < 0.003 across seeds).

H. Additional Ablations

We conduct four additional ablations beyond those in Section 4.4: rollback depth (how many tokens are discarded per event), basis size K, the entropy gate contribution, and the choice of ℓ_{crit}. Together these experiments decompose LPSR's $+15.2$ pp gain over Standard AR into the contributions of each design decision.

H.1. Effect of Rollback Depth

Table 11 ablates the number of tokens rolled back. Rolling back exactly 1 token (current implementation) is optimal; larger rollbacks overfit to the steering direction and reduce accuracy.

Table 11. MATH-500 accuracy vs. rollback depth on 100-problem validation split.

Rollback depth	0 (no rollback)	1 (LPSR)	2	3
Accuracy	0.288	**0.443**	0.418	0.391

H.2. Basis Vector Count

Reproduced in Table 2 in Appendix C.

H.3. LPSR Without Entropy Gate

Setting $\tau_H = 0$ disables the entropy gate (the cosine gate alone triggers rollback). The result on MATH-500:

$$\text{Acc (no entropy gate)} \;=\; 0.390 \qquad \text{vs.} \qquad \text{Acc (full LPSR)} \;=\; 0.440, \tag{31}$$

a drop of 5.0 pp. The entropy gate is essential for filtering false-positive phase shifts during near-deterministic structural token generation.

H.4. LPSR at $\ell_{\text{crit}} = 14$ vs. 16

The full MATH-500 results are:

$$\text{Acc}(\ell_{\text{crit}} = 14) \;=\; 0.292, \qquad \text{AUC}(\ell_{\text{crit}} = 14) \;=\; 0.718, \tag{32}$$

$$\text{Acc}(\ell_{\text{crit}} = 16) \;=\; 0.440, \qquad \text{AUC}(\ell_{\text{crit}} = 16) \;=\; 0.652. \tag{33}$$

The accuracy gap is:

$$\Delta_{\text{acc}} \;=\; 0.292 - 0.440 \;=\; -0.148 \quad (-14.8\text{ pp}), \tag{34}$$

despite the detection AUC being 0.066 higher at layer 14. This *detection–correction dissociation* confirms that $\ell_{\text{crit}} = 16$ is the correct operating point. Full discussion in Section 5.1.

I. Broader Impact and Limitations

Broader impact. LPSR improves the reliability of mathematical reasoning in LLMs without requiring fine-tuning or additional training data. Potential beneficial applications include educational AI tutors, automated theorem assistance, and scientific computation verification. We do not foresee direct harmful applications; the method does not enhance deception or any targeted harm.

Limitations (extended).

1. **Domain transfer.** The basis $\mathcal{V}$ is calibrated on MATH-500 training data. Deployment on coding, logical reasoning, or natural language tasks would require domain-specific calibration. A preliminary experiment on 50 HumanEval problems showed only marginal improvement (+2.1%), suggesting the basis is not immediately transferable. We note that deploying LPSR in high-stakes settings (medical, legal) without domain-specific calibration carries risk—the basis $\mathcal{V}$ is tuned for mathematical errors and may not generalise to safety-critical error types.
2. **Model scale.** We tested primarily at 8B. Preliminary experiments at 70B (with the steering basis recalibrated) are ongoing. Larger models may have more distributed error representations, reducing the effectiveness of single-layer monitoring.
3. **Compute overhead.** On problems with many rollbacks, the overhead can reach $10\times$ standard AR. A budget constraint mechanism (maximum rollbacks per problem) is straightforward to add but was not explored here.
4. **Layer choice sensitivity.** The strong dependence on ℓ_{crit} (Table 7) means that a poorly-chosen monitoring layer can be catastrophic. Future work should explore multi-layer monitoring or learned layer selection.

J. Connection to "Thinking" Depth and Reasoning Complexity

Recent work (Merrill & Sabharwal, 2024; Valmeekam et al., 2023) discusses LLM reasoning as a form of bounded computation. LPSR can be viewed through this lens: the phase-shift detector measures whether the model's "internal thought process" is coherent. The rollback-with-injection mechanism is analogous to backtracking in symbolic search, applied at the representation level.

The finding that rollbacks concentrate at 53–58% into generation (Figure 5) parallels observations in Guo et al. (2025) ("thinking" models tend to exhibit a "reflective pause" at roughly the midpoint of their reasoning trace). LPSR makes this implicit dynamic explicit and actionable: rather than waiting for a wrong final answer, it detects and corrects the pivotal moment when the reasoning trajectory diverges.

An implication is that the "illusion of thinking" (Shojaee et al., 2025)—where models appear to reason but are actually performing shallow pattern matching—may be detectable via phase shifts. If a model's residual stream does not exhibit coherent directional flow at ℓ_{crit}, its apparent chain-of-thought may not correspond to genuine computation. This hypothesis connects LPSR's detection mechanism to broader questions about the nature of LLM reasoning, and we leave its formal investigation to future work.