REWIND: Unlocking the Latent Reasoning Capabilities of LLMs during Test-Time Decoding

Anonymous ACL submission

Abstract

Large Language Models possess remarkable reasoning capabilities, yet standard autoregressive decoding is fundamentally passive: once a token is generated, the model cannot revisit it, allowing localized errors to cascade into complete reasoning failures. While existing test-time scaling methods attempt to address this through iterative refinement, they often rely on random exploration, which can spend refinement budget on regions that are not responsible for the error. We introduce REWIND (Reasoning Error Windback via INtrinsic Detection), a training-free inference-time mechanism that transforms decoding into an active refinement process. Our key insight is that reasoning divergences often produce identifiable *entropy spikes*, sharp increases in predictive uncertainty that precede downstream failure. REWIND uses these spikes as proxies for the onset of instability, backtracks to a nearby coherent boundary, regenerates the suffix, and filters proposals with an MH-style accept step. We evaluate REWIND across specialist (Qwen2.5-Math), generalist (Tulu-3), and reasoning-distilled (DeepSeek-R1) models and observe improvements, with the clearest gains on harder mathematical benchmarks. On the AIME2025 benchmark, our method improves the DeepSeek-R1-Distill-1.5B model from 13.33% to 26.67% in our controlled evaluation and pushes the generalist Tulu-3-8B to surpass 90% accuracy on GSM8K, suggesting that targeted revision can elicit capabilities that standard decoding leaves unused.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable reasoning capabilities across diverse domains, ranging from complex mathematical problem-solving to multi-step logical inference [7; 33; 8]. However, autoregressive decoding is inherently passive: models generate tokens sequentially, committing to each step without retrospective revision [11]. Despite this latent reasoning capacity, the rigid process creates a fragile inference dynamic. Once a reasoning trajectory derails because of an isolated hallucinated premise or a computational error, the model cannot self-correct [16]; instead, it compounds errors, producing coherent but incorrect conclusions [30; 5]. This failure mode indicates that the limitation often lies not in the model's knowledge, but in the absence of mechanisms to arrest and reverse inference deviations, as illustrated in Figure 1. Intervening before errors cascade requires a reliable signal of impending failure.

Fortunately, such signals exist. A substantial body of literature has established indicators for measuring internal confusion in LLMs [2; 26]. Metrics like Shannon entropy are widely employed to *detect* hallucinations, flag uncertain generations, or trigger post-hoc filtering. However, these signals remain largely observational: they are typically used to discard or rerank completed outputs instead of intervening during generation. This reveals a critical gap: *we can diagnose reasoning failures in real time, yet standard decoding algorithms lack the agency to act on this diagnosis.*

In this work, we propose REWIND (Reasoning Error Windback via INtrinsic Detection), a test-time mechanism that transforms decoding from a passive process into an active refinement loop. Our key insight is that reasoning errors are not always instantaneous events but *processes*: an early instability can trigger a cascade of increasing confusion, culminating in uncertainty. We operationalize this insight by distinguishing between the *symptom*, the moment of peak entropy where the model is overtly confused, and an earlier entropy-gradient spike that serves as an observable proxy for the onset of the divergence.

Unlike methods that rely on blind parallel sampling [32] or random exploration [19], which can spend computation on already-correct reasoning

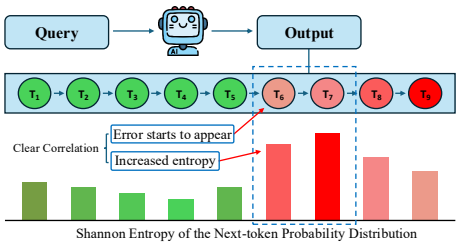

Figure 1: Schematic illustration of autoregressive irreversibility. Standard decoding commits to each token step-by-step. The **red trajectory** indicates a reasoning error; because the model lacks a mechanism to "rewind," this local error can propagate into an incorrect final answer. We propose a controller that revises from an earlier, more stable prefix.

steps, our approach backtracks to a likely onset of instability and regenerates from semantically coherent boundaries. By targeting an earlier distributional shift rather than only the later entropy peak, we enable the model to "rewind and rethink" at critical junctions without external supervision or model retraining.

Extensive experiments validate this approach across diverse model families. In our controlled AIME2025 evaluation, our method improves DeepSeek-R1-Distill-1.5B from 13.3% to 26.7%, and it pushes Tulu-3-8B beyond 90% accuracy on GSM8K. These results suggest that some useful reasoning trajectories remain accessible through targeted test-time refinement.

Our contributions are summarized as follows:

- **Entropy Dynamics as an Actionable Signal:** We establish empirically that local entropy dynamics, specifically the distinction between peak uncertainty and earlier entropy-gradient spikes, serve as useful proxies for reasoning divergence. This transforms entropy from a passive diagnostic into an active control signal for targeted refinement.

- **A Training-Free Refinement Framework:** We present a general inference-time controller that upgrades passive decoding to an active, error-aware process. By decoupling the *trigger signal* (entropy dynamics) from the *revision action* (semantic-aligned backtracking), our framework enables targeted correction without model retraining.

- **Empirical Gains on Challenging Benchmarks:** We demonstrate improvements across specialist (Qwen2.5-Math), generalist (Tulu-3), and reasoning-distilled (DeepSeek-R1) models, with the clearest gains on harder mathematical reasoning tasks and controlled comparisons against DoLa, Self-Refine, and standard Power Sampling.

2 Related Work

Training-Free Test-Time Scaling. Scaling laws have traditionally focused on pre-training parameters and data [18]. However, recent work shifts attention to *"test-time compute"*, showing that allocating additional inference-time budget can substantially improve reasoning performance without changing model weights [7]. A widely adopted strategy is *Parallel Sampling and Voting*, exemplified by Self-Consistency [32], which aggregates multiple independent reasoning paths and selects the most consistent outcome. While effective, this "generate-then-select" paradigm is expensive and offers limited recourse once an error has propagated through a sampled path. Alternatively, *Tree Search* methods such as Tree of Thoughts (ToT) prompting [37] and LLM-MCTS [40; 23] explicitly explore the reasoning space. Although powerful, they typically rely on task-specific heuristics or value functions, limiting their generalizability. Most relevant to our work are *MCMC-based Refinement* techniques. Power Sampling [19] uses Metropolis–Hastings-style correction to favor higher-scoring sequences under a sharpened distribution. Similarly, other iterative refinement methods [35] progressively improve a draft through repeated proposals. However, these methods typically employ uninformed proposal mechanisms that rely on random re-sampling windows, which can make refinement less targeted under a fixed compute budget. Our work advances this line by replacing blind proposals with an entropy-guided, boundary-aligned revision mechanism.

Uncertainty-Aware Decoding and Reasoning. Quantifying and managing predictive uncertainty is critical to reliable reasoning in LLMs. Uncertainty-aware decoding methods typically leverage distributional statistics to distinguish confident continuations from potentially erroneous trajectories. Various proxies for uncertainty have driven novel decoding strategies. For instance, *DoLa* [9] contrasts logits between early and late layers, utilizing Jensen–Shannon divergence to amplify factual knowledge. *Typical Decoding* [27]

constrains generated tokens to have information content close to the conditional entropy; the commonly used Typical-p sampler is a truncated implementation of this local typicality principle rather than a separate uncertainty metric. Furthermore, uncertainty can also serve as a control signal for efficiency; *Adaptive Consistency* [1] dynamically allocates compute, reserving expensive sampling paths only for high-uncertainty queries.

Among these metrics, Shannon entropy [29] remains a fundamental and widely adopted indicator of intrinsic model uncertainty. High entropy has been shown to correlate strongly with hallucinations [25] and generation quality [22]. Recent works refine this granular signal to capture "semantic uncertainty"; for example, Farquhar et al. [12] and Kuhn et al. [20] cluster semantically equivalent generations to estimate hallucination risk while reducing sensitivity to surface phrasing. In multi-step reasoning, entropy dynamics has also been used to actively guide exploration, steering models away from uncertain trajectories [39; 17].

From Passive Detection to Active Intervention. A key limitation of prior work is that uncertainty is often used passively [15; 24]. Many approaches treat entropy as a diagnostic for *error detection*—e.g., re-ranking outputs, filtering uncertain generations, or early stopping—rather than as a trigger for *error correction* [38; 6]. Huang et al. [16] argue that LLMs struggle to self-correct partly because they fail to reliably locate the earlier mistake that causes downstream errors. Our work bridges this gap by turning uncertainty from an observational metric into an actionable control signal. In particular, we show that sharp entropy increases (spikes) can localize the onset of reasoning divergence, enabling targeted backtracking and boundary-aligned regeneration.

3 Method

In this section, we present REWIND (Figure 2). We first review power sampling for large language models and then describe our targeted revision mechanism for mitigating reasoning failures efficiently. Let L_{context} denote the number of prompt tokens; all uncertainty statistics below are computed only over generated tokens after this fixed context.

3.1 Preliminaries: Power Sampling for LLMs

Let $p(x_t|x_{<t})$ denote the conditional probability distribution over the vocabulary $\mathcal{V}$ defined by an LLM, where $x_{<t}$ represents the context tokens. Standard autoregressive generation samples each token $x_t \sim p(\cdot|x_{<t})$. Although effective for general fluency, this approach treats all generative steps equally, potentially accumulating errors in tasks that require multi-step reasoning.

Power Sampling Karan and Du [19] addresses this by reshaping the distribution to favor high-confidence tokens. The target *power distribution* is defined as:

$$\pi(x) \propto p(x)^\alpha = \prod_{t=1}^{T} p(x_t|x_{<t})^\alpha, \qquad (1)$$

where $\alpha \geq 1$ acts as an inverse temperature parameter (equivalent to $1/\tau$). Since the normalization constant of $\pi(x)$ is intractable, sampling is performed via Markov Chain Monte Carlo (MCMC). Specifically, given a current sequence $\mathbf{x}$ and a proposal sequence $\mathbf{x}'$ generated by a proposal distribution $q(\mathbf{x}'|\mathbf{x})$, the Metropolis-Hastings (MH) acceptance ratio $A(\mathbf{x}'|\mathbf{x})$ is:

$$A(\mathbf{x}'|\mathbf{x}) = \min\left(1, \frac{\pi(\mathbf{x}') \cdot q(\mathbf{x}|\mathbf{x}')}{\pi(\mathbf{x}) \cdot q(\mathbf{x}'|\mathbf{x})}\right). \qquad (2)$$

Typically, the base LLM itself serves as the proposal mechanism (i.e., $q(\cdot)$ follows the standard autoregressive distribution).

3.2 Reasoning Error Windback via INtrinsic Detection

While Power Sampling improves generation quality via iterative refinement, its standard implementation deploys random revision, which is computationally inefficient. In reasoning tasks, model errors are often preceded by detectable fluctuations in the model's internal confidence state.

Thus, we propose Reasoning Error Windback via INtrinsic Detection (REWIND), a general framework that allocates computational resources to regions of high instability. Let $\mathcal{M} : \mathbb{R}^{|\mathcal{V}|} \to \mathbb{R}$ be a diagnostic function that maps the token probability distribution $p(\cdot|x_{<t})$ to a scalar uncertainty metric M_t. Our approach proceeds in four phases: localization, backtracking, boundary-aligned regeneration, and MH-style acceptance.

Phase 1: Symptom Detection (Localization). For a generated sequence $\mathbf{x}$, we first compute the uncertainty metric trajectory $\mathbf{M} = \{M_1, \ldots, M_T\}$.

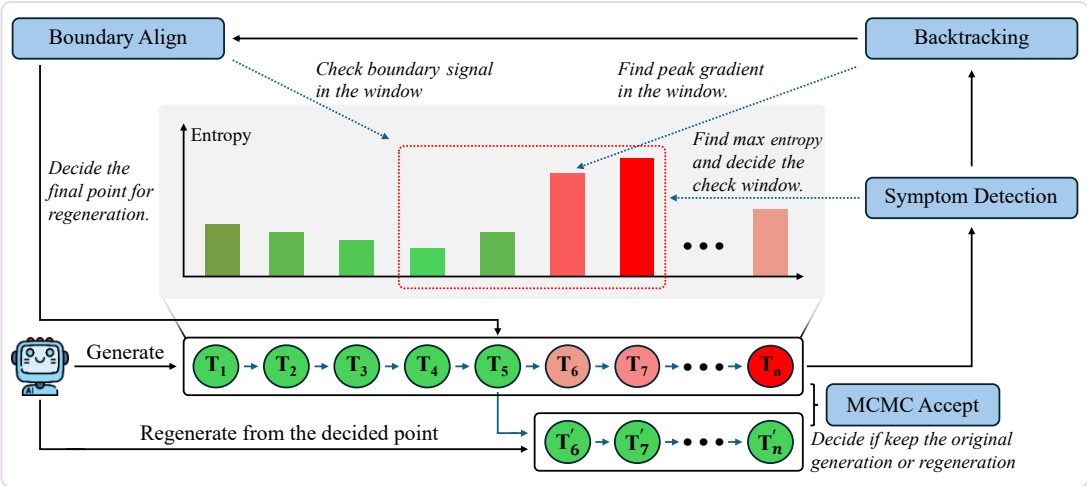

Figure 2: Illustration of REWIND. We illustrate how REWIND transforms passive decoding into an active refinement process. While standard methods might intervene at the *symptom* (maximum entropy, t^*), our mechanism backtracks to an earlier entropy-gradient spike (t_{start}), an observable proxy for the onset of distributional instability. By aligning the intervention with semantic boundaries (t_{aligned}) and applying an MH-style acceptance filter, we regenerate the suffix from a cleaner prefix and retain proposals that improve the sequence score.

We identify the position of maximum metric intensity t^*, which represents the "symptom" where the model is most visibly confused:

$$t^* = \arg\max_{t > L_{\text{context}}} M_t. \tag{3}$$

This position t^* serves as the anchor for potential reasoning failures.

Phase 2: Onset Localization (Backtracking). The high uncertainty at t^* is often merely the manifestation of a divergence that occurred earlier. Simply revising from the peak t^* is insufficient. Instead, we look for the *onset* of the instability. We define a *revision window* of size w and compute the discrete gradient (rate of change) of the metric:

$$\Delta M_t = M_t - M_{t-1}. \tag{4}$$

We locate the starting position t_{start} where the metric sharply increases (the "spike") within the window preceding the peak:

$$t_{\text{start}} = \arg\max_{t \in [\max(L_{\text{context}}+1, t^*-w),\, t^*]} \Delta M_t. \tag{5}$$

This heuristic identifies a likely *onset* of distributional instability rather than treating the maximum-uncertainty token itself as the error location. We use a fixed local window rather than scaling w with the full sequence length: the controller is intended to find the local onset immediately preceding the uncertainty peak, not to search the entire chain of thought. Figure 3 studies this locality assumption and shows that $w = 5$ performs best in our MATH500 sweep.

While our framework is compatible with various distributional statistics (e.g., Kurtosis, Skewness, Top-K Similarity as analyzed in Appendix E), we select Shannon Entropy as the default instantiation for M_t:

$$M_t = H(p) = -\sum_{v \in \mathcal{V}} p(v|x_{<t}) \log p(v|x_{<t}). \tag{6}$$

We prioritize entropy because it integrates probability mass across the full distribution rather than only the leading mode. This is useful when the model becomes torn between incompatible continuations: probability mass can flatten even when no single token confidence statistic is decisive. Appendix E compares entropy against several distributional alternatives, and Appendix G adds maximum-probability, probability-gap, and varentropy checks for common confidence-style baselines.

Phase 3: Semantic Boundary Alignment. Arbitrary truncation at t_{start} can disrupt syntactic or semantic coherence (e.g., breaking a number or a function name). To maintain coherence, we introduce the *Semantic Boundary Alignment*. We search backwards from t_{start} for the nearest natural delimiter (e.g., line breaks '\n', punctuation, or step markers):

$$t_{\text{aligned}} = \max\{t \le t_{\text{start}} \mid x_t \in \mathcal{S}_{\text{bound}}\}, \tag{7}$$

where $\mathcal{S}_{\text{bound}}$ is a predefined set of boundary tokens. This set is intentionally lightweight rather than a language-specific parser; it includes common separators that appear in our English mathematical chain-of-thought format. If no boundary is found within a look-back limit k, we default to $t_{\text{aligned}} = t_{\text{start}}$. The revision then proceeds by keeping the prefix $\mathbf{x}_{1:t_{\text{aligned}}-1}$ and resampling the suffix. Appendix D ablates a restricted delimiter set, Appendix G reports an expanded-set check, and we treat multilingual or non-LaTeX boundary detection as an implementation limitation rather than a theoretical requirement.

Phase 4: MH-Style Acceptance Criterion. Let $\mathbf{x}$ be the original sequence and $\mathbf{x}'$ be the proposal where tokens after t_{aligned} are regenerated. By construction, the two sequences share the same prefix $\mathbf{x}_{<t_{\text{aligned}}} = \mathbf{x}'_{<t_{\text{aligned}}}$, so the implemented score only compares the regenerated suffixes:

$$\log r_{\text{suffix}} = S(\mathbf{x}', t_{\text{aligned}}) - S(\mathbf{x}, t_{\text{aligned}}),$$
$$S(\mathbf{y}, a) = \sum_{t=a}^{|\mathbf{y}|} \begin{bmatrix} \alpha \log p(y_t \mid y_{<t}) \\ -\log q_{\text{LM}}(y_t \mid y_{<t}) \end{bmatrix}. \quad (8)$$

The proposal is accepted if $\log u < \log r_{\text{suffix}}$, where $u \sim \mathcal{U}(0, 1)$. Because REWIND uses a state-dependent entropy-guided choice of t_{aligned}, this filter should be read as an MH-style accept criterion rather than a proof of an exact reversible Markov chain. A fully exact MH kernel would also include the probability of selecting the intervention site under both forward and reverse states. In this work, the acceptance step is used as a practical quality filter for targeted proposals; Appendix H reports acceptance and repair/regression behavior.

Although each refinement step selects a single intervention point, the proposal is not a single-token edit: it regenerates the entire suffix after t_{aligned}. Thus, when a later mistake is induced by an earlier erroneous prefix, one accepted proposal can remove multiple downstream errors at once. If an independent error remains in the regenerated continuation, subsequent refinement iterations recompute the entropy trajectory and can target a different location. We provide an empirical coverage analysis in Appendix I.

3.3 Theoretical Intuition

The controller in Sec. 3.2 relies on a simple mechanistic picture: when the model entertains incompatible continuations, probability mass spreads across alternatives and entropy rises. Under a latent-mixture view $p(x \mid s_t) = \sum_z p(z \mid s_t) p(x \mid s_t, z)$, the predictive entropy decomposes into within-interpretation uncertainty and cross-interpretation disagreement, $H_t = A_t + B_t$, where $B_t = I(X_t; Z \mid s_t)$. A sharp increase in B_t therefore produces an entropy-gradient spike near the onset of distributional instability, while the later entropy peak is often only a downstream symptom. We do not interpret this signal as a proof of token-level causality; rather, it gives a tractable criterion for selecting candidate revision points. Appendix A formalizes sufficient conditions under which this local gradient identifies the onset within the searched window; the result is conditional and is meant as a mechanistic explanation rather than a guarantee for all reasoning errors.

4 Experiments

4.1 Experimental Setup

We evaluate the effectiveness of our REWIND controller across three diverse model families to assess robustness and generality. Specifically, we use Qwen2.5-Math-7B [36] as a representative specialist math model, Tulu-3-Llama-8B [21] as a strong generalist instruction-tuned baseline, and DeepSeek-R1-Distill-1.5B [13] to assess performance on a lightweight reasoning-distilled model known for advanced chain-of-thought capabilities.

We report single-shot accuracy on five benchmarks spanning varying difficulty levels: GSM8K (grade-school math) [10], MATH500 (competition math)[14], GPQA (graduate-level reasoning) [28], AIME2025 (frontier math competition) [4; 3], and BBH (broad logical reasoning) [31]. The main comparison in Table 1 reports the mean over three independent runs with different random seeds. To check robustness under common evaluation protocols and broader settings, Appendix K additionally reports results under standard few-shot CoT prompting, Appendix B reports compute-matched self-consistency comparisons, Appendix L reports larger-model and code-generation results, and Appendix M reports standard deviations for representative Qwen2.5-Math settings.

Baselines. We compare our method against a comprehensive suite of decoding strategies, grouped into three categories. First, *Distribution-based* methods include Vanilla, Temperature Sampling ($T = 0.25$), and Typical-p [27]. Second,

Revision-based techniques include DoLa [9] (layer-contrastive decoding) and Self-Refine [16], which iteratively regenerates and edits the response based on the model's own feedback. Third, we compare with Power Sampling [19] as the representative baseline of MCMC.

Implementation Details & Reproducibility. For Power Sampling and our MH-style refinement controller, we use the same refinement budget to ensure fair and reproducible comparisons. We set the target power distribution to $\alpha = 4$ and run $K = 10$ refinement iterations per instance. For Boundary Alignment, we use a small set of heuristic delimiters $\mathcal{S}_{\text{bound}} = \{\texttt{\textbackslash n}, \texttt{.}, \texttt{,}, \texttt{\$}, \texttt{)}\}$ and choose the cut point by scanning backward from the detected entropy spike to the nearest token in $\mathcal{S}_{\text{bound}}$. All experiments are run on NVIDIA H800 GPUs with the vLLM inference engine. Our code is available anonymously at https://anonymous.4open.science/r/emnlp-rewind-D310.

4.2 Main Results

Averaged over three random seeds, our entropy-guided active controller improves accuracy across model families and benchmarks, with the clearest gains on harder mathematical tasks where localized errors can derail long derivations. Table 1 reports the full comparison against standard decoding baselines and inference-time methods such as DoLa, Self-Refine, and MCMC. We emphasize the larger and more stable gaps, while treating small differences on low-sample or saturated benchmarks as directional evidence.

The clearest gains appear on challenging mathematical benchmarks. For Qwen2.5-Math-7B, REWIND improves MATH500 from 66.80% under temperature sampling to 77.60%, and from 72.60% under standard MCMC to 77.60% under a comparable iterative refinement budget. On DeepSeek-R1-Distill-1.5B, REWIND reaches 26.67% on AIME2025 compared with 16.67% for MCMC in the same controlled setting. We interpret these gains as evidence that targeted proposals can recover trajectories missed by random resampling. The result is not a claim that REWIND is an exact reversible sampler for the same target distribution; rather, it improves the proposal mechanism while using an MH-style score filter.

The broader pattern is more nuanced on saturated or low-sample settings. For Tulu-3-Llama-8B, REWIND improves GSM8K from 86.27% to 90.90%, but gains on BBH and GPQA are smaller and should be read as supportive rather than decisive. AIME2025 contains only 30 questions, so we pair the headline numbers with the multi-seed analysis in Appendix M. Additional robustness is reported in Appendix B and Appendix K.

Table 1: **Main results.** Single-shot accuracy (%) across five benchmarks, averaged over three runs. The **best** and second-best scores are highlighted. Vanilla denotes decoding with temperature $T = 1$.

Methods	GSM8K	MATH500	GPQA	AIME2025	BBH
Qwen2.5-Math-7B (Specialist)					
Vanilla	61.40	48.80	26.77	1.67	63.80
Temperature	75.86	66.80	34.59	7.78	71.20
Typical-p	78.26	67.40	31.31	7.78	74.40
Self-refine	77.64	68.40	32.82	6.67	69.60
DoLa	77.85	70.20	34.84	10.00	73.60
MCMC	79.45	72.60	35.35	10.00	75.20
REWIND	**84.31**	**77.60**	**35.86**	**16.67**	**77.20**
Tulu-3-Llama-8B (Generalist)					
Vanilla	86.27	44.20	24.74	1.11	56.20
Temperature	88.17	44.80	25.25	1.67	57.00
Typical-p	87.67	46.20	28.78	1.67	54.80
Self-refine	87.87	45.60	24.24	1.67	55.40
DoLa	88.32	45.00	25.76	1.67	57.40
MCMC	89.08	50.40	27.27	2.22	57.80
REWIND	**90.90**	**52.00**	**29.79**	**3.33**	**59.40**
DeepSeek-R1-Distill-1.5B (Reasoning)					
Vanilla	63.84	55.40	27.27	13.33	60.40
Temperature	67.94	57.00	28.79	13.33	62.80
Typical-p	66.56	59.20	29.80	14.44	65.20
Self-refine	68.52	55.80	27.77	13.33	62.00
DoLa	71.65	58.40	29.80	14.44	65.60
MCMC	73.70	61.20	29.29	16.67	64.00
REWIND	**77.80**	**64.40**	**31.81**	**26.67**	**66.80**

4.3 Mechanism Analysis

Our REWIND is built on two distinct design choices: (1) backtracking from the uncertainty peak to an earlier entropy-gradient spike (onset localization) and (2) aligning revisions with coarse semantic boundaries. To test whether these choices contribute beyond generic regeneration, Table 2 reports an ablation study on the MATH500 dataset using the Qwen2.5-Math model.

Validating the Onset-Localization Hypothesis. A central pattern is that Peak Revision (74.20%) underperforms our backtracking approach (77.60%). While revising at the peak is still better than global random sampling (72.60%), it does not identify the most effective intervention point. This supports

Table 2: **Mechanism ablation on MATH500.** Accuracy (%) under a matched refinement budget.

Strategy	Localization	Alignment	Acc.
Standard MCMC	Global random	None	72.60
Random at delimiters	Global random	Natural delimiters	73.20
Peak revision	Local peak	Natural delimiters	74.20
Full w/o delimiters	Gradient spike window	None	75.10
Random in window	Local peak window	Natural delimiters	75.20
Restricted delimiters	Gradient spike window	. and \n only	76.80
REWIND (ours)	**Gradient spike window**	**Full delimiters**	**77.60**

Table 3: **Peak vs. onset localization.** Accuracy (%) on MATH500 under naive peak revision and gradient tracing.

Metric	Naive	Tracing	Δ Gain
MCMC	72.6	-	-
Shannon Entropy	74.2	**77.6**	**+3.4**
KL Divergence (Temporal)	71.8	75.0	+3.2
KL Divergence (Uniform)	73.4	74.8	+1.4
Top-K Similarity ($K = 5$)	73.4	74.2	+0.8
Top-K Similarity ($K = 3$)	72.6	73.4	+0.8
Skewness	73.2	73.4	+0.2
Kurtosis	72.0	73.0	+1.0

our hypothesis that the moment of peak uncertainty (t^*) is often a symptom, whereas a more useful proposal point occurs earlier near the entropy-gradient spike. By the time entropy peaks, the model may have already conditioned on several questionable tokens; backtracking to the spike gives the proposal a cleaner prefix.

The Role of Boundary Alignment. Removing Semantic Boundary Alignment leads to a -2.5% performance degradation. Randomly intervening at delimiters (73.20%) also falls short, indicating that boundary alignment alone is insufficient. The stronger Random-in-Window baseline (75.20%) shows that localizing the search region helps, but REWIND still gains +2.4 points by selecting the entropy-gradient spike rather than a random point in that window. The restricted-delimiter variant, which keeps only . and \n, remains close to the full delimiter set (76.80% vs. 77.60%), suggesting that coarse boundary alignment helps preserve coherence but the exact hand-written separator set is not the primary source of the gain. Together, these results suggest that both targeted localization and coarse boundary-aligned truncation contribute to the final gain.

Appendix H further analyzes whether entropy spikes correlate with failed trajectories and whether interventions tend to repair or regress answers. Appendix I analyzes how suffix regeneration covers multi-error chains across MCMC iterations.

Robustness of the Gradient Tracing Strategy. To test whether the onset-localization pattern is specific to Shannon entropy, we decouple the *localization strategy* from the *metric choice*. On MATH500, we compare two revision strategies across multiple distributional statistics (KL divergence, Top-K similarity, skewness, and kurtosis). Naive Peak Detection revises at the position with maximum metric value ($t^* = \arg\max M_t$), implicitly assuming the error coincides with the point of highest confusion. In contrast, **Targeted Gradient Tracing (Tracing)** revises at the maximum metric gradient within a preceding window ($t_{\text{start}} = \arg\max \Delta M_t$), treating the sharp distributional shift as a proxy for an earlier useful intervention point.

The results in Table 3 show a consistent pattern: across all tested metrics, the Targeted strategy outperforms the Naive strategy. The **+3.4** point gap between Peak Revision and our backtracking approach is the clearest evidence in this setting that the entropy peak is often a downstream symptom and the earlier spike is a better intervention proxy. Smaller gains for noisier metrics, such as *Kurtosis* (+1.0) and *Skewness* (+0.2), should be interpreted cautiously, but they point in the same direction. Moreover, the strategy appears robust to the choice of signal: while Shannon entropy yields the largest improvement, *KL Temporal* also achieves a substantial boost (+3.2) when paired with Gradient Tracing. Importantly, Table 3 isolates the localization strategy on MATH500 only; it does not imply that every trigger metric generalizes equally well to harder benchmarks. This suggests that the controller's gains are driven by both *where* it intervenes and which uncertainty signal is used.

Table 4 shows that entropy is the strongest trigger among the main multi-benchmark signals tested here. KL Divergence (Uniform) is also consistently positive, but with smaller gains. The AIME2025 column highlights that simpler peakedness-style heuristics can be brittle on frontier problems, motivating our use of full-distribution entropy as the default signal.

Window-Size Sensitivity. The revise window w controls how far the controller searches backward from the uncertainty peak. Figure 3 shows an

Table 4: **Trigger-signal robustness.** Accuracy (%) when different distributional metrics are used to select revision points.

Trigger metric (M_t)	GSM8K	MATH500	AIME2025
MCMC	79.45	72.60	10.00
Shannon Entropy (Ours)	**84.08**	**77.60**	**15.83**
KL Divergence (Uniform)	81.27	74.80	13.33
Skewness	80.97	73.40	12.22
Top-K Similarity ($K = 3$)	76.19	73.40	12.22
KL Divergence (Temporal)	77.86	75.00	10.00
Kurtosis	80.59	73.00	8.89
Top-K Similarity ($K = 5$)	79.91	74.20	7.78

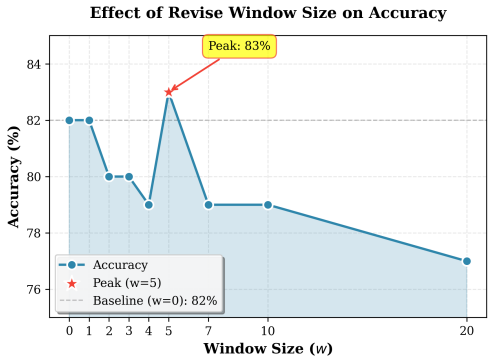

Figure 3: **Effect of revise window size (w).** Accuracy on MATH500 peaks at $w = 5$, supporting a local-window controller.

inverted-U trend on MATH500: performance peaks at $w = 5$, while larger windows degrade as the search begins to select unrelated fluctuations. This supports treating the entropy-gradient spike as a local onset signal rather than a reason to backtrack over the entire chain of thought.

Additional Robustness. Appendix E provides additional discussion of trigger-signal behavior, including why the Top-K results in Tables 3 and 4 answer different questions. Appendix F shows that the same controller also improves a temperature-scaled sequence score.

5 Discussion

Our results suggest that test-time scaling need not only increase the number of independent samples. A complementary direction is to allocate compute *within* a trajectory, using internal uncertainty to decide where the model should reconsider. This perspective separates REWIND from self-consistency, which aggregates complete trajectories, and from random MCMC proposals, which spend compute without identifying where the trajectory became unstable.

The broader implication is that reasoning quality can improve through finer control over the temporal structure of generation. When failures are localized, revising the suffix from a cleaner prefix can be more targeted than restarting generation or voting over many full samples. This framing positions REWIND as a trajectory-repair method rather than a general replacement for sampling-and-voting. In this sense, REWIND treats inference-time compute as an adaptive control budget over an evolving reasoning trace, rather than only as a multiplier on the number of sampled answers.

Design principle. REWIND follows a simple decomposition: detect where the trajectory becomes unstable, intervene at a coherent boundary before the instability propagates, and accept revisions only when they pass a sequence-level score filter. This separates detection, intervention, and acceptance into modular components. Entropy is the default diagnostic in our implementation, but the framework could incorporate stronger uncertainty signals, task-aware boundary detectors, or alternative scoring objectives without changing the overall control loop.

When it helps. The empirical pattern suggests that trajectory-level control is most useful when errors are localized and downstream tokens depend strongly on an early unstable step. This helps explain the larger gains on hard math dataset, where a single flawed derivation step can derail the solution, compared with saturated settings where many independent samples already converge to the same answer. Conversely, when the model is confidently wrong or majority voting is already stable, targeted revision has less room to improve.

6 Conclusion

In this work, we addressed the inherent fragility of autoregressive decoding, where localized errors often cascade into catastrophic reasoning failures. We introduced REWIND, a framework that transforms inference from a passive generation process into an active, score-filtered revision loop. Our empirical results suggest that standard decoding leaves some reasoning capability unused, especially when localized errors derail otherwise plausible trajectories. REWIND points toward a finer-grained form of inference-time scaling: not only generating more answers, but deciding when and where a single reasoning trajectory should be revised.

Limitations

Computational Cost. The primary trade-off of our proposed framework is the increased computational overhead associated with iterative refinement. Unlike standard autoregressive decoding which generates the sequence in a single pass, refinement-based methods require repeated sampling and scoring. Empirically, we observe that the average token consumption of standard MCMC Power Sampling is approximately **8.84**× that of Temperature Sampling. Our REWIND, due to its active backtracking mechanism and block-level regeneration, incurs a slightly higher cost of **9.26**×. This cost means REWIND is not a replacement for cheap single-pass decoding or self-consistency on saturated benchmarks; its compute-accuracy trade-off is most attractive on harder, low-consensus settings where targeted revision improves individual trajectories.

References

[1] Pranjal Aggarwal, Aman Madaan, Yiming Yang, and 1 others. 2023. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. *arXiv preprint arXiv:2305.11860*.

[2] Riad Ahmed Anonto, Md Labid Al Nahiyan, and Md Tanvir Hassan. 2025. When safety blocks sense: Measuring semantic confusion in llm refusals. *arXiv preprint arXiv:2512.01037*.

[3] Art of Problem Solving. 2025. 2025 aime 2. https://artofproblemsolving.com/wiki/index.php/2025_AIME_II. Accessed: 2026.

[4] Art of Problem Solving. 2025. 2025 aime i. https://artofproblemsolving.com/wiki/index.php/2025_AIME_I. Accessed: 2026.

[5] Sebastian Barros. 2025. I think, therefore i hallucinate: Minds, machines, and the art of being wrong. *arXiv preprint arXiv:2503.05806*.

[6] Mohammad Beigi, Sijia Wang, Ying Shen, Zihao Lin, Adithya Kulkarni, Jianfeng He, Feng Chen, Ming Jin, Jin-Hee Cho, Dawei Zhou, and 1 others. 2024. Rethinking the uncertainty: a critical review and analysis in the era of large language models. *arXiv preprint arXiv:2410.20199*.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[8] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, and 1 others. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

[9] Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

[10] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

[11] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, and 1 others. 2023. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36:70293–70332.

[12] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

[13] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

[14] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

[15] David Herrera-Poyatos, Carlos Peláez-González, Cristina Zuheros, Andrés Herrera-Poyatos, Virilo Tejedor, Francisco Herrera, and Rosana Montes. 2025. An overview of model uncertainty and variability in llm-based sentiment analysis: challenges, mitigation strategies, and the role of explainability. *Frontiers in Artificial Intelligence*, 8:1609097.

[16] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.

[17] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

[18] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei.

2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

[19] Aayush Karan and Yilun Du. 2025. Reasoning with sampling: Your base model is smarter than you think. *arXiv preprint arXiv:2510.14901*.

[20] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

[21] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

[22] Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

[23] Lihui Liu. 2025. Monte carlo tree search for graph reasoning in large language model agents. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 4966–4970.

[24] Huan Ma, Jingdong Chen, Joey Tianyi Zhou, Guangyu Wang, and Changqing Zhang. 2025. Estimating llm uncertainty with evidence. *arXiv preprint arXiv:2502.00290*.

[25] Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 9004–9017.

[26] Kelly Marchisio, Wei-Yin Ko, Alexandre Bérard, Théo Dehaze, and Sebastian Ruder. 2024. Understanding and mitigating language confusion in llms. *arXiv preprint arXiv:2406.20052*.

[27] Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121.

[28] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

[29] Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

[30] Adi Simhi, Itay Itzhak, Fazl Barez, Gabriel Stanovsky, and Yonatan Belinkov. 2025. Trust me, i'm wrong: High-certainty hallucinations in llms. *arXiv preprint arXiv:2502.12964*.

[31] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

[32] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

[33] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

[34] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

[35] Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P Lillicrap, Kenji Kawaguchi, and Michael Shieh. 2024. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*.

[36] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

[37] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

[38] Fanghua Ye, Mingming Yang, Jianhui Pang, Longyue Wang, Derek Wong, Emine Yilmaz, Shuming Shi, and Zhaopeng Tu. 2024. Benchmarking llms via uncertainty quantification. *Advances in Neural Information Processing Systems*, 37:15356–15385.

[39] Jinghan Zhang, Xiting Wang, Fengran Mo, Yeyang Zhou, Wanfu Gao, and Kunpeng Liu. 2025. Entropy-based exploration conduction for multi-step reasoning. *arXiv preprint arXiv:2503.15848*.

[40] Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2023. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*.

Appendix Overview

The appendix provides supporting material for the main empirical and methodological claims. Appendix A gives a conditional theoretical account for entropy-gradient localization, while Appendix O explains the sequence-level motivation behind the accept/reject refinement loop. Appendix B compares REWIND with self-consistency under similar inference budgets, and Appendix C compares targeted refinement with a larger single-pass model. Appendices D–G report robustness checks for boundary alignment, trigger signals, supplementary controller variants, and alternative scoring objectives; the main text reports window-size sensitivity in Figure 3. Appendices H–J analyze entropy diagnostics, repair/regression behavior, multi-error coverage, and residual failures. Finally, Appendix K, Appendix L, and Appendix M provide additional prompting, generalization, and stability results, followed by qualitative case studies in Appendix N and the full algorithm in Appendix P.

A Theoretical Justification

In this appendix, we provide a conditional theoretical account for REWIND. The goal is not to prove that every reasoning error is entropy-detectable, but to explain a regime in which entropy and its gradient are principled signals for detecting the onset of reasoning divergence. The key insight is that some reasoning failures manifest as *semantic fragmentation*: the model becomes uncertain between conflicting interpretations, causing mutual information to spike. We formalize this intuition and derive sufficient conditions under which the entropy gradient identifies the local onset of fragmentation.

A.1 Setup and Notation

Let $s_t := x_{<t}$ denote the prefix at step t. The next-token entropy is defined as

$$H_t := H(X_t \mid s_t) = -\sum_v p(v \mid s_t) \log p(v \mid s_t), \tag{9}$$

and the entropy gradient is $\Delta H_t := H_t - H_{t-1}$.

A.2 Main Result

Theorem A.1 (Entropy Gradient Localizes Semantic Fragmentation)**.** *Assume the model's predictive distribution admits a latent decomposition*

$$p(x \mid s_t) = \sum_{z \in \mathcal{Z}} p(z \mid s_t) \cdot p(x \mid s_t, z), \tag{10}$$

where z represents implicit semantic interpretations. Define the within-world entropy $A_t := \mathbb{E}_Z[H(X_t \mid s_t, Z)]$ and the cross-world disagreement $B_t := I(X_t; Z \mid s_t)$.

(Decomposition) *The entropy decomposes as*

$$H_t = A_t + B_t, \quad \textit{hence} \quad \Delta H_t = \Delta A_t + \Delta B_t. \tag{11}$$

(Localization) *If semantic fragmentation occurs at time t_0, causing a mutual information jump $\Delta B_{t_0} \geq \Gamma$, and the within-world entropy is locally smooth with $|\Delta A_t| \leq L$ over the searched neighborhood $\mathcal{W}(t^*)$, then*

$$\Delta H_{t_0} \geq \Gamma - L. \tag{12}$$

(Identification) *If additionally $\Delta B_t \leq \gamma$ for all competing positions $t \in \mathcal{W}(t^*)$, $t \neq t_0$ (no comparable MI jumps in the local search window) and $\Gamma - \gamma > 2L$, then*

$$t_0 = \arg\max_{t \in \mathcal{W}(t^*)} \Delta H_t. \tag{13}$$

Proof. **Part 1** follows from the definition of conditional mutual information. Recall that $I(X_t; Z \mid s_t) = H(X_t \mid s_t) - H(X_t \mid s_t, Z)$. Rearranging gives $H(X_t \mid s_t) = H(X_t \mid s_t, Z) + I(X_t; Z \mid s_t)$. Since $H(X_t \mid s_t, Z) = \sum_z p(z \mid s_t) H(X_t \mid s_t, z) = \mathbb{E}_Z[H(X_t \mid s_t, Z)] = A_t$ and $I(X_t; Z \mid s_t) = B_t$, we obtain $H_t = A_t + B_t$.

Part 2: At t_0, we have $\Delta H_{t_0} = \Delta A_{t_0} + \Delta B_{t_0}$. By the smoothness assumption, $\Delta A_{t_0} \geq -L$. By the fragmentation assumption, $\Delta B_{t_0} \geq \Gamma$. Thus $\Delta H_{t_0} \geq -L + \Gamma = \Gamma - L$.

Part 3: For any competing position $t \in \mathcal{W}(t^*)$, $t \neq t_0$, we have $\Delta A_t \leq L$ (local smoothness) and $\Delta B_t \leq \gamma$ (no comparable MI jumps). Thus $\Delta H_t = \Delta A_t + \Delta B_t \leq L + \gamma$. The condition $\Gamma - L > L + \gamma$ (equivalently, $\Gamma - \gamma > 2L$) ensures $\Delta H_{t_0} \geq \Gamma - L > L + \gamma \geq \Delta H_t$ for all such t. □

Remark A.2 (Interpretation of Assumptions). The theorem requires three components. First, a *structural assumption*: the latent mixture decomposition exists, which aligns with Bayesian interpretations of in-context learning where latent concepts govern generation [34]. Second, a *local smoothness condition*: within any fixed interpretation z, the model's uncertainty should not fluctuate as sharply as the fragmentation signal inside the small window searched by the controller. This is weaker than assuming globally smooth entropy over an entire long chain of thought; abrupt but task-legitimate

phase changes elsewhere can violate the assumption without invalidating the local argument. Third, a *signal strength requirement*: the fragmentation event produces a detectably large MI jump relative to background fluctuations. The identification statement is conditional; it holds when the local signal-to-noise ratio $(\Gamma - \gamma)/L$ is sufficiently large.

A.3 Why Backtracking to Gradient Peak Outperforms Peak-Based Revision

Proposition A.3 (Clean Prefix Advantage)**.** *Let t_0 be the fragmentation onset (gradient peak) and $t^* > t_0$ be the entropy peak. Assume that once fragmentation occurs, the cross-world disagreement remains elevated, i.e., $B_t \geq B_{t_0-1} + \Gamma'$ for some $\Gamma' > 0$ and all $t \in [t_0, t^*]$. Here Γ' denotes the post-onset elevation level; it need not equal the jump size Γ in Theorem A.1, although in the simple persistent-jump case they are of the same order.*

Backtracking to t_0 retains the prefix $x_{<t_0}$, which was generated under coherent conditions where $\Delta B_t \leq \gamma$ for all $t < t_0$. In contrast, backtracking to t^* retains the prefix $x_{<t^*}$, which includes tokens from $[t_0, t^*)$ that were generated under elevated cross-world disagreement. Since the prefix $x_{<t_0}$ does not condition on any tokens generated during fragmentation, regeneration from this clean prefix has higher probability of producing a coherent continuation.

Proposition A.4 (Heuristic Proposal Quality Improvement)**.** *Under the setup of Theorem A.1 and Proposition A.3, let t_0 be the fragmentation onset and $t^* > t_0$ be the entropy peak. If regeneration from a clean prefix produces coherent output with probability $(1 - \rho)$, and if expected token-level negative log-likelihood tracks the local entropy level, then the expected score advantage of gradient-based over peak-based revision is approximately lower bounded by*

$$\begin{aligned} &\mathbb{E}[\log p(x'_{\text{grad}})] - \mathbb{E}[\log p(x'_{\text{peak}})] \\ &\quad \geq (1-\rho)(t^* - t_0)(\Gamma - \gamma). \end{aligned} \tag{14}$$

Proof Sketch. The key observation is that gradient-based backtracking retains a *clean* prefix (generated under coherent conditions with $B_t \leq \gamma$), while peak-based backtracking retains a *contaminated* prefix (containing tokens generated under fragmentation with $B_t \geq \Gamma$).

Consider the interval $[t_0, t^*)$. Under gradient-based revision, these tokens are regenerated from a clean prefix; with probability $(1 - \rho)$, regeneration maintains coherence, so their expected uncertainty remains near the low-disagreement regime $A_t + \gamma$. Under peak-based revision, these tokens are retained from the original fragmented generation, whose uncertainty is in the elevated-disagreement regime, written here as $A_t + \Gamma$ for the simple persistent-jump case.

The expected log-probability difference over this interval is therefore

$$\begin{aligned} &(1-\rho) \sum_{t=t_0}^{t^*-1} [(A_t + \Gamma) - (A_t + \gamma)] \\ &\quad = (1-\rho)(t^* - t_0)(\Gamma - \gamma). \end{aligned} \tag{15}$$

This argument uses the identity $\mathbb{E}_{X_t \sim p(\cdot|s_t)}[-\log p(X_t \mid s_t)] = H_t$ as a local proxy for expected next-token NLL. It is therefore a heuristic score comparison, not a trajectory-level proof for arbitrary realized tokens: after accepting or rejecting a token, the future state changes and entropies do not add linearly in general. The purpose of the proposition is to formalize why a cleaner prefix should improve proposal quality under the fragmentation model. □

A.4 When Does Fragmentation Produce a Detectable Signal?

Lemma A.5 (MI Lower Bound)**.** *Suppose at step t, two semantic worlds z_1, z_2 have total mass $m = p(z_1 \mid s_t) + p(z_2 \mid s_t)$, normalized conditional weights $\tilde{\pi}_1, \tilde{\pi}_2 \geq \eta$ within this two-world subset, and $\mathrm{TV}(p(\cdot \mid s_t, z_1), p(\cdot \mid s_t, z_2)) \geq \delta$ (they disagree on predictions). Then*

$$I(X_t; Z \mid s_t) \geq 2m\,\eta(1-\eta)\delta^2. \tag{16}$$

Proof Sketch. Let $p_i = p(\cdot \mid s_t, z_i)$ denote the conditional distributions. We first condition on the event $Z \in \{z_1, z_2\}$, whose probability is m, and write the normalized weights as $\tilde{\pi}_i = p(z_i \mid s_t)/m$. For the two-component conditional mixture $\bar{p}_{12} = \tilde{\pi}_1 p_1 + \tilde{\pi}_2 p_2$, the total variation between each component and the conditional mixture satisfies $\mathrm{TV}(p_1, \bar{p}_{12}) = \tilde{\pi}_2 \mathrm{TV}(p_1, p_2) \geq \tilde{\pi}_2 \delta$ and similarly for p_2.

Applying Pinsker's inequality $D_{\mathrm{KL}}(p \| q) \geq 2\,\mathrm{TV}(p, q)^2$ within this conditional two-world mixture yields a conditional mutual-information lower bound of $2\tilde{\pi}_1\tilde{\pi}_2\delta^2$. Multiplying by the subset probability m gives the stated contribution to the full mutual information by the chain rule and non-negativity of the remaining components.

Since $\tilde{\pi}_1, \tilde{\pi}_2 \geq \eta$ and $\tilde{\pi}_1 + \tilde{\pi}_2 = 1$, we have $\tilde{\pi}_1\tilde{\pi}_2 \geq \eta(1-\eta)$, yielding $I(X_t; Z \mid s_t) \geq 2m\eta(1-\eta)\delta^2$. □

Remark A.6. This lemma shows that fragmentation produces a detectable signal when the model is "torn" between interpretations (η not too small) that make substantially different predictions (δ not too small). The quadratic dependence on δ reflects that small prediction differences yield negligible MI, while substantial disagreement creates a strong signal.

A.5 Scope and Limitations

The theoretical statements are **conditional**: they apply to errors that manifest as semantic fragmentation, where the model becomes uncertain about which interpretation to follow. The framework does not cover two important failure modes. First, *confident errors*, where the model commits to a wrong but internally coherent interpretation, produce no entropy spike and thus cannot be detected by our method. Second, *gradual drift*, where errors accumulate slowly without a sharp transition, violates the assumption of a localized MI jump.

The signal-to-noise condition $\Gamma - \gamma > 2L$ provides insight into when REWIND is most effective. Complex reasoning tasks can produce stronger fragmentation signals (larger Γ) when errors occur, while the local-window condition controls the background variation L. This helps explain why our method yields larger gains on challenging benchmarks such as AIME2025 and MATH500 compared to simpler tasks, but it also identifies a failure mode: if legitimate step-to-step uncertainty changes are as large as the fragmentation signal, the gradient heuristic can mislocalize the revision point.

Summary. The theoretical framework rests on four components: (1) the latent decomposition provides the structural foundation, (2) the local within-world smoothness parameter L characterizes the background noise level inside the controller's search window, (3) the MI jump Γ at t_0 quantifies the fragmentation signal strength, and (4) the condition $\Gamma - \gamma > 2L$ ensures local identifiability.

The core insight is that the entropy gradient ΔH_t can capture the *onset* of semantic fragmentation when the mutual information term $I(X_t; Z \mid s_t)$ jumps sharply at t_0, while the within-world term A_t evolves smoothly. This decomposition explains why backtracking to the gradient peak can outperform revision at the entropy peak: the former regenerates from a cleaner prefix before downstream tokens have been conditioned on the unstable segment, while the latter retains more contaminated context.

Table 5: **Compute-matched comparison with self-consistency.** Accuracy (%) is reported for Qwen2.5-Math-7B. SC@10 uses roughly 10× sampling compute, while REWIND uses roughly 9.26× token cost.

Benchmark	SC@1	SC@10	REWIND
AIME2025	9.67	13.3	**16.7**
MATH500	66.8	77.2	**77.6**
GSM8K	75.9	**84.5**	84.3

Table 6: **Scaling self-consistency on AIME2025.** REWIND remains above the SC curve at a comparable compute budget.

Approach	Compute budget	AIME2025 Acc.
SC@1	1.0×	9.67
SC@3	3.0×	12.0
SC@7	7.0×	14.3
SC@10	10.0×	13.3
REWIND	$\sim$ 9.26×	**16.7**

B Compute-Matched Self-Consistency

Since REWIND consumes more tokens than single-pass decoding, we compare it against self-consistency (SC) under similar inference budgets. Table 5 reports SC@10 using the same Qwen2.5-Math-7B backbone. SC@1 denotes one sampled trajectory from the SC sweep, not the deterministic Vanilla row in Table 1. REWIND is most advantageous on AIME2025, where answers are diverse and majority voting is less stable. On MATH500, the two methods are close, and on GSM8K SC@10 is slightly stronger, suggesting that voting is highly effective when the model can already generate many consistent correct trajectories.

We further inspect the vote concentration of SC@10. The average number of winning votes is 3.13 on AIME2025, 7.40 on MATH500, and 7.67 on GSM8K. This indicates that SC becomes less stable on harder problems where independent samples disperse across many answers. In contrast, REWIND directly revises an individual trajectory and is therefore not dependent on answer-level consensus. Table 6 expands the AIME2025 comparison across several SC budgets.

Table 7: **Combining REWIND with self-consistency.** REWIND-SC@3 applies self-consistency over three REWIND-refined trajectories.

Benchmark	SC@30	REWIND-SC@3
MATH500	81.4	**83.4**
GSM8K	90.4	**91.2**

Finally, SC and REWIND are orthogonal. SC is an aggregation wrapper over multiple independent samples, while REWIND is a trajectory-level refinement algorithm. Table 7 shows that combining the two can outperform a larger pure-SC budget.

C Comparison with Qwen2.5-Math-72B

We additionally compare our final Qwen2.5-Math-7B setting with a much larger Qwen2.5-Math-72B single-pass baseline. This comparison is not compute matched: REWIND deliberately spends additional inference time on iterative revision, while the 72B result uses one sampled trajectory at temperature $T = 0.25$. The purpose is to test a different trade-off: whether targeted test-time refinement can exchange extra decoding time for accuracy gains that would otherwise require substantially larger model scale. Table 8 reports this comparison across all five benchmarks.

The 72B model remains stronger on the saturated GSM8K benchmark, but REWIND surpasses this single-pass 72B baseline on four of the five benchmarks: MATH500, GPQA, AIME2025, and BBH. This supports the view that our final method is best understood as a test-time scaling strategy: it spends more inference time to revise unstable trajectories, and in difficult or low-consensus settings this extra computation can recover accuracy that is not obtained by a single larger-model sample.

D Boundary Alignment Robustness

The boundary set used by REWIND is intentionally lightweight: it avoids cutting through obviously incomplete spans while adding no additional parsing cost. To test whether the exact hand-written delimiter set drives the gains, we restrict the set to only two general-purpose anchors, `\n` and `.`. Table 9 shows that this restricted variant preserves most of the full method's performance on both MATH500 and HumanEval. This suggests that the main gain comes from entropy-gradient localization, while boundary alignment mainly improves syntactic coherence of the regenerated suffix.

E Signal Robustness

Our active controller can be instantiated with any distributional statistic M_t as the trigger signal. To evaluate the choice of Shannon entropy, we compare it against higher-order moments, top-K similarity, and KL-based alternatives across GSM8K, MATH500, and AIME2025.

In the main signal comparison, Shannon entropy yields the largest gains over MCMC on all three benchmarks: +4.63 points on GSM8K, +5.00 on MATH500, and +5.83 on AIME2025. KL Divergence (Uniform) is also consistently positive, but with smaller gains. The pattern supports entropy as a strong default signal, while also showing that the framework is not tied to entropy alone.

The harder AIME2025 setting exposes a failure mode for simpler heuristics. Top-K similarity and kurtosis are reasonable on easier benchmarks but can degrade performance on AIME2025. Top-$K5$, for example, lowers AIME2025 accuracy relative to MCMC. This is not inconsistent with Table 3: that table tests whether gradient tracing improves peak localization on MATH500, whereas Table 4 tests whether each metric is a reliable trigger across datasets. Confidence-style signals can conflate confidently wrong states with genuinely stable reasoning, causing the controller to intervene at the wrong time or miss useful revision points.

F Score Robustness

The main text reports window-size sensitivity in Figure 3. Here, we test whether the active controller depends on the specific geometry of Power Sampling.

We apply the same targeted proposal and MH-style filtering procedure using a temperature-scaled score induced by standard temperature sampling ($T = 0.25$).

Table 10 shows that entropy-guided proposals remain useful when the scoring objective changes from a power score to a temperature-scaled score. This supports interpreting REWIND as a practical proposal-refinement operator rather than a technique tied to one exact target distribution.

G Supplementary Controller Robustness

We run additional single-seed checks to stress two implementation choices that are not exhaustively covered in the main tables: the choice of trigger metric and the granularity of boundary/window selection. These experiments are intended as diagnos-

Table 8: **Comparison with Qwen2.5-Math-72B.** Accuracy (%) is reported across five benchmarks. The 72B baseline is single-pass sampling with $T = 0.25$, while REWIND uses the Qwen2.5-Math-7B backbone with iterative refinement.

Method	GSM8K	MATH500	GPQA	AIME2025	BBH
Qwen2.5-Math-72B, single-pass ($T = 0.25$)	**90.52**	75.40	33.33	6.67	67.60
REWIND (Qwen2.5-Math-7B)	84.31	**77.60**	**35.86**	**16.67**	**77.20**

Table 9: **Robustness to restricted delimiters.** The restricted setting keeps only `\n` and `.`; full uses `\n`, `.`, `,`, `$`, and `)`.

Benchmark	MCMC	Restricted	Full
MATH500	72.60	76.80	**77.60**
HumanEval	56.10	57.90	**58.50**

Table 10: **Robustness across scoring objectives.** We apply REWIND to the temperature-scaled sequence score with $T = 0.25$. Even when the scoring objective changes, the active controller improves reasoning accuracy over baseline sampling.

Metric	MATH500	GPQA	AIME2025
Temperature	69.20	30.82	7.50
+REWIND	71.00	35.23	**13.33**
Gain	+1.80	+4.41	**+5.83**

tic supplements to the main multi-seed comparison, not as replacements for it.

Table 11 shows that mode-focused confidence signals do not consistently replace entropy. Maximum probability obtains a higher single-seed score on AIME2025, but AIME2025 contains only 30 questions and the same metric falls below the MCMC baseline on MATH500. Probability gap and varentropy also trail entropy on MATH500. We therefore treat these confidence-style signals as noisy diagnostics rather than stable default triggers.

Table 12 suggests that increasing the controller's flexibility is not automatically beneficial. Expanding the delimiter set slightly weakens performance, consistent with the risk of introducing noisy cut points. A simple adaptive enlargement of the look-back range also underperforms the fixed local window. These results support our default choice of a lightweight delimiter set and $w = 5$ as a local time-scale parameter for the evaluated reasoning traces, while leaving more sophisticated length-aware controllers to future work.

H Entropy Signal Diagnostics

We further examine whether entropy anomalies are associated with failed reasoning trajectories. Since the benchmarks do not provide token-level annotations of the first incorrect reasoning step, we use final-answer correctness as an observable proxy and analyze token-level entropy trajectories from Qwen2.5-Math-7B on AIME2025 across multiple random seeds. This diagnostic is not intended to be a perfect confusion matrix for latent reasoning errors; rather, it tests whether the unsupervised signal used by REWIND is predictive of downstream failure.

Table 13 shows that incorrect trajectories tend to have higher entropy peaks and sharper local gradient spikes. This supports the use of entropy dynamics as an unsupervised proxy for likely reasoning instability, while also indicating that the signal is imperfect and should be paired with an acceptance criterion rather than used as a hard overwrite rule.

We also measure the observable consequence of interventions at the answer level. As shown in Table 14, REWIND fixes 46 trajectories that standard MCMC answers incorrectly while regressing 21 trajectories that standard MCMC answers correctly. This favorable repair-to-regression ratio suggests that the entropy-guided proposal is biased toward useful edits. Moreover, REWIND does not automatically accept every proposed regeneration: the MH-style acceptance step rejects proposals with low acceptance probability, which helps protect correct trajectories when a high-entropy position is a false positive.

Acceptance rate alone does not measure proposal quality: random MCMC can accept lateral moves that do not change the final answer. We therefore interpret Table 15 together with Table 14. The lower acceptance rate of REWIND, combined with a positive repair-to-regression ratio, suggests that the acceptance filter rejects a portion of harmful or low-value regenerations while retaining enough targeted edits to improve final accuracy.

Table 11: **Supplementary trigger-metric checks.** Max probability, probability gap, and varentropy are added as common confidence-style baselines under the same MCMC revision budget. Entropy values are repeated from Table 4 for reference.

Trigger metric (M_t)	**MATH500**	**AIME2025**
MCMC	72.60	10.00
Shannon Entropy (main)	**77.60**	15.83
Max Probability	70.60	**16.67**
Probability Gap	71.80	13.33
Varentropy	70.60	13.33

Table 12: **Supplementary boundary and window checks.** Expanded delimiters add additional punctuation and closing brackets to the default set. The adaptive-window variant uses the full delimiter set and allows a larger look-back window capped at 20 tokens.

Setting	**MATH500**	**AIME2025**
Full delimiters, fixed $w = 5$ (main)	**77.60**	**15.83**
Expanded delimiters, fixed $w = 5$	74.00	13.33
Full delimiters, adaptive window (max 20)	73.60	10.00

Table 13: **Entropy diagnostics on AIME2025.** Incorrect trajectories exhibit larger peak entropy and sharper local entropy-gradient spikes. Both signals have non-trivial zero-shot discriminative power for final-answer failure.

Signal	**Correct traj.**	**Incorrect traj.**	**AUROC**
Peak entropy	3.00	3.33	0.71
Max entropy gradient	2.49	2.79	0.65

Table 14: **Answer-level repair and regression analysis on MATH500.** Relative to standard MCMC, REWIND produces more corrections than regressions, suggesting that targeted intervention is not merely perturbing correct solutions.

Setting	**Fixed errors**	**Regressions**	**Repair / regression**
Qwen2.5-Math-7B on MATH500	46	21	2.19×

Table 15: **Proposal acceptance ratios.** REWIND has slightly lower acceptance than random MCMC proposals, suggesting that its targeted proposals are filtered more selectively by the MH-style acceptance criterion.

Benchmark	**MCMC**	**REWIND**	Δ
AIME2025	44.4	39.4	-5.0
MATH500	57.3	54.1	-3.2
GSM8K	65.6	63.4	-2.2

Table 16: **Iterative coverage across refinement rounds on MATH500.** The controller revisits multiple distinct intervention points over the course of the refinement loop, consistent with suffix regeneration covering multiple downstream errors.

Setting	**MCMC iterations**	**Distinct positions / problem**
Qwen2.5-Math-7B on MATH500	10	3.18

I Iterative Multi-Error Coverage

Although REWIND selects one aligned intervention point per refinement proposal, each accepted proposal regenerates the entire suffix after that point. This means that a single successful rewrite can remove multiple downstream mistakes when they are consequences of one upstream divergence. To check whether the controller remains stuck on the same site, we inspect the aligned backtracking positions visited across MATH500 runs. Table 16 summarizes this coverage statistic: on average, one problem traverses **3.18** distinct sequence positions over $K = 10$ iterations, suggesting that repeated proposals often sweep over different locations rather than repeatedly revising a single token neighborhood.

The key point is not that REWIND edits multiple locations simultaneously, but that suffix regeneration naturally removes many downstream errors once an upstream unstable segment is revised. When an independent mistake remains in the new continuation, later refinement rounds can still identify a fresh entropy spike and revise a different segment.

Table 17: **Failure taxonomy on MATH500 (manual inspection).** Confident errors are a minority of all evaluated problems; most residual failures occur after the entropy spike is detected but regeneration still does not recover a correct continuation.

Failure type	Share of all problems	Share of failures
Confident errors	7.6%	33.9%
Detected but unrecovered	14.8%	66.1%

J Residual Failure Analysis

To better characterize the remaining failures, we manually inspect the failed trajectories on MATH500. Because the benchmark does not annotate the first incorrect reasoning token, we use a coarse taxonomy based on the observed entropy trace and the outcome of regeneration. Table 17 summarizes the two main failure modes: (i) confident errors, where the model is wrong without a salient entropy spike, and (ii) detected-but-unrecovered cases, where REWIND identifies instability but the regenerated suffix still does not reach a correct answer.

This breakdown suggests that the main limitation is not the existence of a single-point proposal mechanism, but the fact that some wrong trajectories are already overconfident, and some detected instabilities still require a stronger proposal search than the base model can supply.

K Standard Few-Shot CoT Evaluation

Our main experiments use a single-shot setting to isolate the effect of the inference-time controller. To test whether the gains persist under common prompting protocols, we additionally evaluate Qwen2.5-Math-7B with standard 8-shot CoT prompting on the math benchmarks. Table 18 shows that REWIND continues to improve over vanilla decoding, temperature sampling, and standard MCMC under this setting.

L Additional Generalization Results

To probe whether the observed pattern extends beyond the main single-shot setup, we include two supplementary settings. First, Table 19 evaluates a larger specialist model, Qwen2.5-Math-14B, on the same math benchmarks. Second, Table 20 reports HumanEval results for Qwen2.5-Math-7B and Tulu-8B. These results are supplementary rather than the central claim of the paper: our strongest evidence is for mathematical and structured reasoning, while code generation provides an initial check that entropy-guided revision is not restricted to one benchmark family.

M Multi-Seed Stability

We repeat key Qwen2.5-Math-7B experiments over three random seeds. Table 21 reports mean accuracy and standard deviation. The relative ordering between MCMC and REWIND remains stable on both MATH500 and AIME2025.

N Case Studies

We present illustrative examples demonstrating how REWIND identifies and corrects problematic reasoning patterns.

N.1 Case Study 1: Methodological Improvement

Problem: Find the sum of all integer bases $b > 9$ for which 17_b is a divisor of 97_b.

Before Revision (Brute-force Enumeration): The model initially attempts to solve the problem by exhaustively checking each value of k:

> *"For $k = 2$, we get $b(9-2) = 7(2-1)$, which simplifies to $7b = 7$, but this does not satisfy $b > 9$. For $k = 3$, we get $b(9-3) = 7(3-1)$, which simplifies to $6b = 14$, and $b = \frac{14}{6}$, which is not an integer. For $k = 4$... For $k = 5$... For $k = 6$... For $k = 7$... For $k = 8$..."*

This approach is tedious, error-prone, and lacks mathematical elegance.

After Revision (Modular Arithmetic): After detecting high entropy at the beginning of the solution, the model revises to use a more principled approach:

> *"We rewrite $9b + 7$ in terms of $b + 7$:*
>
> $$9b+7 = 9(b+7)-63+7 = 9(b+7)-56$$
>
> *Thus: $9b + 7 \equiv -56 \pmod{b + 7}$. For $b+7$ to divide $9b+7$, it must divide -56. Therefore, $b + 7$ must be a divisor of 56: $\{1, 2, 4, 7, 8, 14, 28, 56\}$."*

Improvement: The revision transforms a brute-force enumeration into an elegant number-theoretic solution, reducing the problem to finding divisors of 56.

Table 18: **Performance under standard 8-shot CoT prompting.** Accuracy (%) is reported for Qwen2.5-Math-7B. REWIND maintains consistent gains under few-shot CoT prompts.

Benchmark	Vanilla	Temperature	Standard MCMC	REWIND
AIME2025	3.3	7.8	10.0	**16.7**
MATH500	51.2	68.4	73.2	**78.8**
GSM8K	74.9	85.5	86.3	**88.4**

Table 19: **Supplementary results for Qwen2.5-Math-14B.** REWIND preserves the same qualitative gains on a larger model.

Benchmark	Vanilla	Temperature	MCMC	REWIND
AIME2025	2.22	3.33	6.67	**10.00**
MATH500	78.20	82.40	85.00	**87.80**
GSM8K	93.71	94.01	94.54	**94.92**

N.2 Case Study 2: Geometric Error Correction

Problem: In triangle ABC, points D, E lie on $\overline{AB}$ and points F, G lie on $\overline{AC}$. Find the area of heptagon $AFNBCEM$.

Before Revision (Coordinate Error):

The model makes a critical error in setting up coordinates:

> *"F is on $\overline{AC}$ such that $AF = 13$ and $FG = 52$, so F is at $(13, 0)$."*

Error: Point F should lie on line AC, not on the x-axis (which represents line AB). This fundamental mistake would propagate through the entire solution.

After Revision (Correct Setup):

The revision correctly establishes the coordinate system:

> *"Determine the total length of side AB: $AB = AD + DE + EB = 4 + 16 + 8 = 28$.*
> *Determine the total length of side AC: $AC = AF + FG + GC = 13 + 52 + 26 = 91$.*
> *Find the coordinates:*
>
> - *D is at $(4, 0)$ on AB*
> - *E is at $(20, 0)$ on AB*
> - *F is at $(0, 13)$ on AC*
> - *G is at $(0, 65)$ on AC*
>
> *"*

Improvement: The revision correctly distinguishes points on AB (along the x-axis) from points on AC (along the y-axis), ensuring geometric consistency.

N.3 Case Study 3: Reasoning Precision

Problem: The plane $x + y + z = 75$ is partitioned by the inequalities $x - yz < y - zx < z - xy$ into three disjoint convex regions. Find the area of the region with finite area.

Before Revision (Vague Approach):

> *"To find the area of this region, we can use the formula for the area of a triangle, which is $\frac{1}{2} \times$ base $\times$ height. However, we need to find the base and height of the triangle that forms the region with finite area."*

The model prematurely assumes the region is triangular without justification.

After Revision (Systematic Analysis):

> *"To find the area of this region, we need to solve the system of inequalities and find the intersection of the plane with the region defined by the inequalities. This involves solving a system of linear inequalities... Once we have the vertices of the region, we can calculate its area using the formula for the area of a polygon."*

Improvement: The revision adopts a more rigorous approach, recognizing the need to first determine the region's vertices before computing its area.

Table 20: **Supplementary HumanEval results.** REWIND also improves code-generation performance on two instruction-tuned models.

Model	Vanilla	Temperature	MCMC	REWIND
Qwen2.5-Math-7B	31.10	52.40	56.10	**58.50**
Tulu-8B	51.22	58.54	62.80	**66.46**

Table 21: **Multi-seed stability on Qwen2.5-Math-7B.** We report accuracy mean $\pm$ standard deviation over three seeds.

Method	MATH500	AIME2025
Vanilla	48.13 ± 0.64	2.22 ± 2.46
Temperature	68.80 ± 0.35	7.78 ± 2.46
MCMC	72.40 ± 1.68	10.00 ± 3.33
REWIND	$\mathbf{77.27} \pm 1.28$	$\mathbf{16.67} \pm 3.33$

N.3.1 Summary

These case studies demonstrate that entropy-guided revision can:

1. **Improve methodology:** Transform inefficient enumeration into elegant mathematical solutions (Case 1)
2. **Correct errors:** Identify and fix fundamental setup mistakes before they propagate (Case 2)
3. **Enhance rigor:** Replace vague reasoning with systematic problem-solving approaches (Case 3)

The entropy signal effectively serves as an "uncertainty detector" that identifies positions where the model's reasoning is most likely to benefit from reconsideration.

O Theoretical Motivation: Sequence-Level Refinement

A natural question arises: *Why use an iterative accept/reject refinement loop instead of simply adjusting the sampling temperature?* In this section, we provide the motivation. Standard temperature sampling is inherently myopic because each token is chosen from a locally normalized distribution. In contrast, sequence-level scoring can evaluate whether a regenerated suffix improves the likelihood of the whole trajectory. Because REWIND uses a state-dependent intervention site, our implementation should be interpreted as an MH-style refinement procedure rather than an exact reversible sampler.

O.1 The Limitation: Local Myopia of Standard Sampling

Standard autoregressive generation controls diversity via a scalar temperature parameter τ. At each time step t, the model samples a token x_t from a *locally* sharpened distribution:

$$p_{\text{local}}(x_t \mid x_{<t}) = \frac{\exp(l_t/\tau)}{\sum_{v \in \mathcal{V}} \exp(l_v/\tau)}, \tag{17}$$

where l denotes the logits. While computationally efficient due to the tractable normalization over the vocabulary $\mathcal{V}$, this process is inherently **myopic**.

The "Early Error" Problem: The model commits to a token x_t based solely on the immediate probability conditioned on the prefix. If a low-probability error occurs early in the sequence (e.g., a hallucinated premise), standard sampling possesses no mechanism to revisit or correct this decision, as the generation is strictly forward-moving. Simply lowering τ merely makes the model more confident in these local choices, potentially reinforcing early errors rather than correcting them.

O.2 The Objective: Sequence-Level Consistency

To address reasoning failures, we score candidate trajectories with a sequence-level objective that prioritizes the **global consistency** of the entire sequence $\mathbf{x}$. One such objective is the Power Distribution [19]:

$$\pi(\mathbf{x}) = \frac{P(\mathbf{x})^\alpha}{Z(\alpha)}, \quad \text{where } Z(\alpha) = \sum_{\mathbf{x}' \in \mathcal{X}} P(\mathbf{x}')^\alpha. \tag{18}$$

Here, $P(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_{<t})$ is the sequence likelihood, and $\alpha > 1$ is the sharpening factor. Unlike local temperature sampling, this score evaluates the sequence holistically. A sequence receives a high score only if the regenerated trajectory is likely under the model across the full suffix, not merely at the next token.

O.3 The Solution: Accept/Reject Refinement

Directly normalizing $\pi(\mathbf{x})$ poses a fundamental challenge: the global partition function $Z(\alpha)$ re-

quires summing over the astronomically large space of all possible sequences $\mathcal{X}$, making it intractable to compute.

The useful insight from Metropolis-Hastings (MH) is that an accept/reject decision can depend on the *ratio* of scores between a proposed revision $\mathbf{x}'$ and the current sequence $\mathbf{x}$. In the exact MH setting, the intractable constant $Z(\alpha)$ cancels out:

$$\begin{aligned} A(\mathbf{x}' \mid \mathbf{x}) &= \min\left(1,\ \frac{\pi(\mathbf{x}')}{\pi(\mathbf{x})}\frac{q(\mathbf{x} \mid \mathbf{x}')}{q(\mathbf{x}' \mid \mathbf{x})}\right) \\ &= \min\left(1,\ \left(\frac{P(\mathbf{x}')}{P(\mathbf{x})}\right)^{\alpha}\frac{q(\mathbf{x} \mid \mathbf{x}')}{q(\mathbf{x}' \mid \mathbf{x})}\right). \end{aligned} \tag{19}$$

Conclusion. REWIND uses this ratio-based view as a practical quality filter for targeted suffix proposals. Since the proposal location is chosen by an entropy-guided, state-dependent controller, we do not claim exact detailed balance for the implemented chain. Instead, the accept/reject step provides a sequence-level safeguard: proposed revisions that lower the score are less likely to replace the current trajectory, while useful targeted rewrites can be retained. This gives standard autoregressive sampling a limited ability to revisit earlier unstable segments without model retraining.

P Algorithm

The complete procedure is summarized in Algorithm 1.

Algorithm 1 Entropy-Guided Targeted Revision

Require: Sequence $\mathbf{x}$, context length L_{ctx}, look-back window w, power α

Ensure: Refined sequence $\mathbf{x}_{\text{final}}$

Phase 1: Symptom Detection (Profiling)
Compute entropy H_t for all tokens x_t where $t > L_{\text{ctx}}$
Identify uncertainty peak: $t^* \leftarrow \arg\max_t H_t$
Phase 2: Onset Localization (Backtracking)
Define search range: $\mathcal{T} \leftarrow [\max(L_{\text{ctx}}, t^* - w), t^*]$
Compute gradients: $\Delta H_t \leftarrow H_t - H_{t-1}$ for $t \in \mathcal{T}$
Locate spike: $t_{\text{start}} \leftarrow \arg\max_{t \in \mathcal{T}} \Delta H_t$
Phase 3: Boundary Alignment
$t_{\text{aligned}} \leftarrow \text{FINDBOUNDARY}(\mathbf{x}, t_{\text{start}})$ ▷ *e.g., nearest '\n'*
Truncate sequence: $\mathbf{x}_{\text{prefix}} \leftarrow \mathbf{x}[1 : t_{\text{aligned}}]$
Phase 4: Refinement & MH-Style Acceptance
Generate suffix: $\mathbf{x}'_{\text{suffix}} \sim P_{\text{LM}}(\cdot \mid \mathbf{x}_{\text{prefix}})$
Construct proposal: $\mathbf{x}' \leftarrow \mathbf{x}_{\text{prefix}} \oplus \mathbf{x}'_{\text{suffix}}$
Compute acceptance ratio
Sample $u \sim \mathcal{U}(0, 1)$
if $\log u < \min(0, \log r)$ **then**
  return $\mathbf{x}'$ ▷ *Accept revision*
else
  return $\mathbf{x}$ ▷ *Reject and keep original*
end if