Dissecting Failure Dynamics in Large Language Model Reasoning

Anonymous ACL submission

Abstract

Large Language Models (LLMs) achieve strong performance through extended inference-time deliberation, yet how their reasoning failures arise remains poorly understood. By analyzing model-generated reasoning trajectories, we find that errors are not uniformly distributed but often originate from a small number of early transition points, after which reasoning remains locally coherent but globally incorrect. These transitions coincide with localized spikes in token-level entropy, and alternative continuations from the same intermediate state can still lead to correct solutions. Based on these observations, we introduce GUARD, a targeted inference-time framework that probes and redirects critical transitions using uncertainty signals. Empirical evaluations across multiple benchmarks confirm that interventions guided by these failure dynamics lead to more reliable reasoning outcomes. Our findings highlight the importance of understanding when and how reasoning first deviates, complementing existing approaches that focus on scaling inference-time computation.

1 Introduction

Large Reasoning Models (LRMs), such as OpenAI o1 (Jaech et al., 2024) and DeepSeek-R1 (Guo et al., 2025), aim to approximate human-like deliberative reasoning by internalizing test-time scaling. Through extended chains of thought, these models decompose complex problems into intermediate steps, enabling multi-stage reasoning and iterative refinement (Wei et al., 2022; Yao et al., 2023; Besta et al., 2024). Reinforcement learning further strengthens this capability by encouraging sustained deliberation on challenging tasks (Uesato et al., 2022; Lightman et al., 2023; He et al., 2025).

Consequently, much recent progress has focused on allocating additional inference-time computation to improve reasoning performance. Representative approaches include generating longer reasoning traces (Snell et al., 2025; Muennighoff et al., 2025), sampling multiple trajectories in parallel (Wang et al., 2023; Snell et al., 2025; Scalena et al., 2025; Xu et al., 2025b), and optimizing inference-time procedures (Zhang et al., 2025a,b). These methods have demonstrated clear gains across benchmarks, reinforcing the view that increased deliberation can be beneficial. Yet these gains provide limited insight into where reasoning goes wrong within a single trajectory, and whether such deviations are isolated events or systematically concentrated in time..

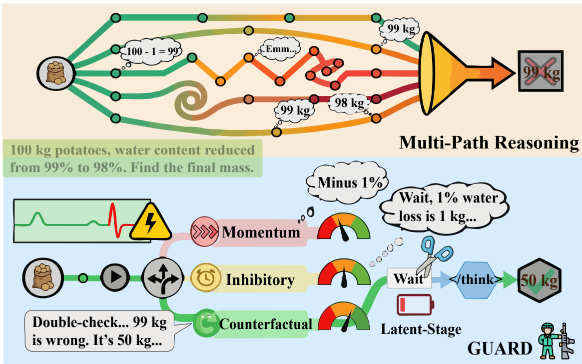

Figure 1: **Comparison of Multi-path Reasoning versus GUARD.** Multi-path reasoning relies on repeated sampling of parallel trajectories, whereas GUARD maintains a single primary trajectory and intervenes only at critical transitions using targeted branching.

In this work, we address this question by analyzing reasoning failures at the trajectory level. Rather than treating incorrect outputs as undifferentiated outcomes, we examine how errors emerge and evolve over time within a single reasoning trace. By systematically analyzing model-generated reasoning trajectories, we study when failures first occur, how they affect subsequent steps, and whether their influence is evenly spread or temporally concentrated.

Our analysis uncovers clear regularities in failure dynamics. Reasoning errors are often temporally concentrated, with failure onsets occurring disproportionately early in the trajectory. After such an onset, the model typically continues with locally coherent but globally incorrect reasoning, allowing early deviations to exert a lasting downstream influence. These critical transitions are marked by localized spikes in token-level entropy, while uncertainty elsewhere remains stable. Moreover, alternative continuations from the same intermediate state can still reach correct solutions, indicating that many failures arise from specific transition choices rather than missing task-relevant knowledge.

Guided by these findings, we introduce **G**uided **U**ncertainty-**A**ware **R**easoning with **D**ecision control (**GUARD**), a lightweight inference-time framework for correcting reasoning trajectories. Rather than expanding computation globally or maintaining multiple parallel paths throughout generation, GUARD follows a single primary reasoning trajectory and introduces only short-horizon local branching when high-risk transitions are detected. These brief interventions allow the model to reconsider critical steps while continuing generation along a single evolving solution. By steering generation away from early deviations and suppressing unproductive late-stage expansion, GUARD improves reasoning reliability without altering the underlying model.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 analyzes recurring failure dynamics in LRM reasoning. Section 4 presents the GUARD framework and its intervention mechanisms. Section 5 evaluates GUARD across multiple benchmarks and model backbones.

2 Related Work

2.1 Large Reasoning Models

The reasoning landscape has shifted from prompt-induced CoT (Wang et al., 2023) to intrinsic Lar ge Reasoning Models (LRMs). Models like OpenAI's o1 (Jaech et al., 2024) and DeepSeek-R1 (Guo et al., 2025) internalize System 2 deliberation, employing latent trajectories optimized via reinforcement learning (Uesato et al., 2022; Lightman et al., 2023; He et al., 2025). While these reasoning-centric models (Guo et al., 2025; Qwen Team, 2025; Team et al., 2024; Abdin et al., 2024) demonstrate strong deliberative capabilities, explicit long-form reasoning remains an inefficient proxy for deliberation, as longer chains frequently increase redundancy without accuracy gains, motivating dynamic inference regulation.

2.2 Test-Time Scaling Strategies

Test-time scaling empowers LLMs to trade inference compute for performance via paradigms ranging from sequential refinement (Shinn et al., 2023; Snell et al., 2025) to parallel sampling (e.g., Best-of-N, Tree-of-Thoughts) and Monte Carlo Tree Search (MCTS) (Wang et al., 2023; Yao et al., 2023; Zhu et al., 2025). However, mainstream methods often rely on blind scaling or expensive verifiers (Wang et al., 2025; Liao et al., 2025), incurring significant redundancy. To mitigate this, recent works leverage intrinsic uncertainty. DTS (Xu et al., 2025b) triggers selective branching based on absolute entropy, while EGB (Li et al., 2025) combines entropy gating with PRMs. Similarly, EAGER (Scalena et al., 2025) and Entro-duction (Zhang et al., 2025a) dynamically reallocate budgets. Yet, these frameworks typically rely on static, non-adaptive thresholds or external verifiers. Crucially, they often generate multiple complete parallel solutions. In contrast, our approach uses an adaptive threshold based on historical entropy percentiles, performing low-budget, in-place interventions on a *single* trajectory without maintaining concurrent hypotheses.

2.3 Efficient Reasoning

Parallel to scaling, extensive work has examined inference efficiency. Chain of Draft (Xu et al., 2025a) was introduced to enforce minimalism, though often at the cost of zero-shot accuracy. Collaborative frameworks (Liao et al., 2025; Chen et al., 2025; Fu et al., 2025; Yang et al., 2025b) offload steps to lighter models but incur alignment complexity and switching overheads. Dynamic strategies like CGRS (Huang et al., 2025), DEER (Yang et al., 2025a), Adaptive Think (Yong et al., 2025), and α1 (Zhang et al., 2025b) modulate depth via confidence or information-theoretic metrics, yet suffer from rigid heuristics or dependencies on pre-computed statistics. Fundamentally, these paradigms prioritize minimizing length, ignoring the algorithmic overhead of control mechanisms and the mining of latent capabilities. In contrast, our approach targets capability maximization with minimal redundancy, repairing fractures rather than merely shortening trajectories.

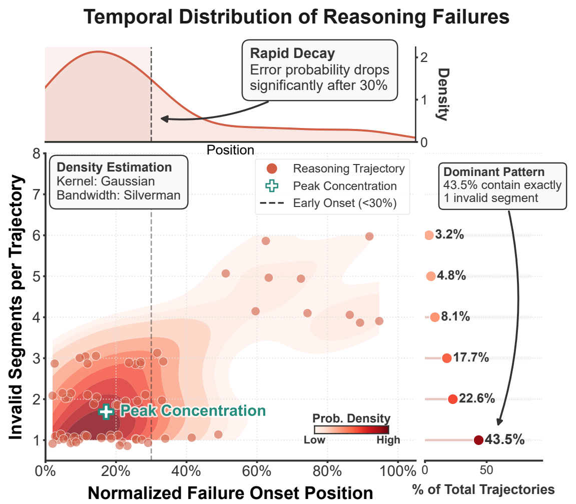

Figure 2: **Early Concentration of Reasoning Failures.** Failure onsets are heavily concentrated in the early stages of generation, and most incorrect trajectories contain only a small number of invalid segments, with 43.5% exhibiting a single error.

3 Empirical Findings on Reasoning Failure Dynamics

In this section, we analyze how reasoning failures arise and propagate along a single generated trajectory. By examining model-produced reasoning traces, we observe several recurring characteristics in how failures develop along the trajectory. Errors often emerge early, expand through subsequent locally coherent steps, exhibit localized uncertainty signatures, and are sometimes recoverable from the same intermediate state. These findings provide a trajectory-level characterization of reasoning failure.

Our analysis is based on reasoning trajectories generated by DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025) on the AMC (AI-MO, 2024) and AIME (MAA Committees, 2025) benchmarks. Each output is segmented into an ordered sequence $\tau = (s_k)_{k=1}^{K}$ using the delimiter \n\n. Segment validity is evaluated using an external oracle based on Gemini 3 Pro (Google DeepMind, 2025), with human verification for quality control. A segment is labeled invalid if it introduces an error that prevents reaching the correct final answer.

3.1 Early Failure Onsets

We begin by examining when reasoning failures arise along a generated trajectory. For each reasoning trace $\tau = (s_k)_{k=1}^{K}$, we use the Oracle to assign a segment-level validity label $\mathcal{O}(s_k) \in \{0, 1\}$ where $\mathcal{O}(s_k) = 1$ indicates that segment s_k is logically valid with respect to the problem context and preceding segments, and $\mathcal{O}(s_k) = 0$ otherwise. We define a *failure onset* at segment s_k whenever $\mathcal{O}(s_{k-1}) = 1 \wedge \mathcal{O}(s_k) = 0$. This definition captures the transition from a valid reasoning prefix to an invalid step.

Figure 2 visualizes the temporal distribution of failure onsets. The top panel shows a strong early concentration, with over 85% of failure onsets occurring within the first 30% of the trajectory. The bottom panel presents the joint distribution of normalized failure onset position and the number of invalid segments per trajectory, estimated using a Gaussian kernel with Silverman bandwidth. The density exhibits a dominant concentration corresponding to early-stage failures accompanied by one to two invalid segments. In particular, 43.5% of trajectories contain exactly one invalid segment.

These patterns indicate that reasoning failures are typically driven by early, localized deviations that account for most errors within a trajectory, rather than by difficulty that accumulates uniformly over time. The concentration of failure onsets in a small number of early segments suggests that the downstream behavior of a trajectory is often determined by a limited set of critical transitions, highlighting the importance of identifying such moments during generation.

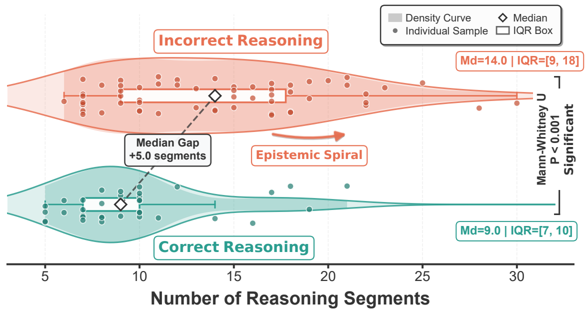

Figure 3: **Segment Count Distribution for Correct and Incorrect Trajectories.** Incorrect trajectories exhibit substantial length expansion following failure onsets.

3.2 Post-Onset Trajectory Expansion

We next examine how the length of a reasoning trajectory relates to its correctness. As shown in Figure 3, incorrect trajectories contain substantially more reasoning segments than correct ones, exhibiting a pronounced long tail in the segment-count distribution.

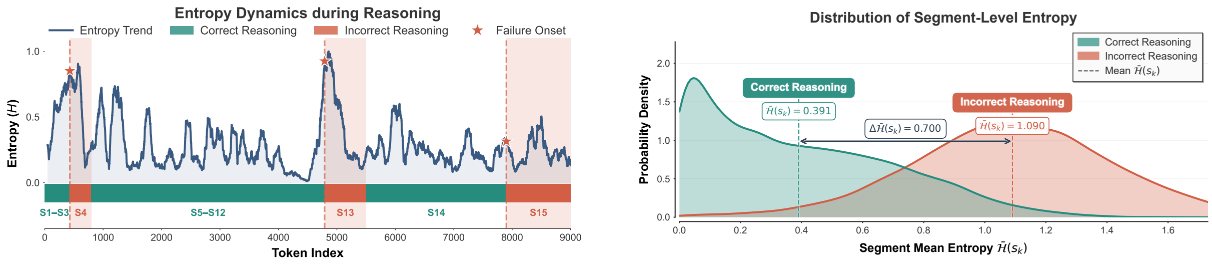

Figure 4: **Left: Entropy aligned to failure onset**, with a localized spike at the transition to invalid reasoning. **Right: Entropy density for valid and invalid segments**, showing higher dispersion and a shifted mean for error segments.

This length expansion occurs predominantly after the failure onset. Section 3.1 shows that most failure onsets arise early in the trajectory, whereas incorrect trajectories continue to generate many additional segments thereafter. Notably, these post-onset segments are not syntactically degenerate or abruptly incoherent. Instead, they form extended sequences of locally plausible reasoning that remain consistent with the initial erroneous premise. We refer to this empirical pattern as an *epistemic spiral*, characterizing the sustained expansion of reasoning following an early failure. Examples of epistemic spiral can be found in Appendix E.

As a result, trajectory length is dominated by post-onset expansion, and extended reasoning is strongly associated with incorrect outcomes, suggesting limited benefit from allocating additional computation to long trajectories.

3.3 Elevated Uncertainty in Error Segments

We next examine whether reasoning errors are accompanied by systematic changes in model uncertainty. Let $\mathbf{z}_t \in \mathbb{R}^{|\mathcal{V}|}$ denote the model logits at token position t. The next-token probability distribution P is defined via the softmax transformation:

$$P(x_t = v \mid x_{<t}) = \frac{\exp(\mathbf{z}_t[v])}{\sum_{v' \in \mathcal{V}} \exp(\mathbf{z}_t[v'])}. \quad (1)$$

We quantify uncertainty using token-wise Shannon entropy $\mathcal{H}(x_t \mid x_{<t})$ and its length-normalized segment aggregation. For a reasoning segment s_k spanning a token subsequence $x_{u_k}, x_{u_k+1}, \ldots, x_{v_k}$, we define:

$$\mathcal{H}(x_t \mid x_{<t}) := -\mathbb{E}_{v \sim P(\cdot \mid x_{<t})} \left[\log P(v \mid x_{<t})\right],$$
$$\bar{\mathcal{H}}(s_k) := \frac{1}{|s_k|} \sum_{t=u_k}^{v_k} \mathcal{H}(x_t \mid x_{<t}). \quad (2)$$

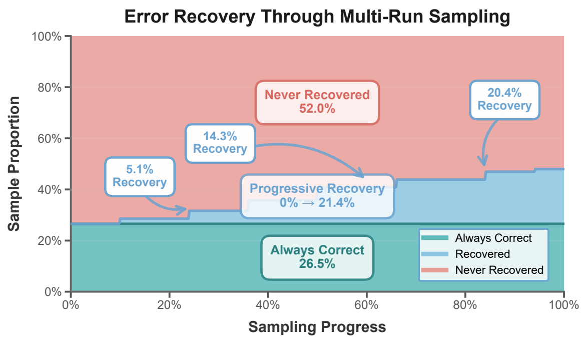

Figure 5: **Recoverability of Reasoning Failures.** Some failures persist across continuations, while others admit correct solutions from the same prefix.

This normalization removes segment-length effects, enabling direct comparison of uncertainty across segments.

We relate these uncertainty measures to the failure onset positions. Figure 4 (left) shows pronounced *local entropy spikes* at failure onsets, where segments corresponding to the onset exhibit a sharp increase in $\bar{\mathcal{H}}(s_k)$ relative to nearby segments. Figure 4 (right) further shows a *global entropy increase* for invalid segments compared to valid ones. Valid segments concentrate in a low-entropy regime, whereas invalid segments form a long-tailed distribution with a significantly higher mean uncertainty ($p < 0.001$).

These results show that uncertainty changes are tightly coupled to where errors arise. Elevated segment entropy marks brief transitions associated with the onset of failure and remains higher in subsequent invalid segments, providing a consistent signal that distinguishes erroneous reasoning from valid progression.

3.4 Local Recoverability of Failures

We next examine whether reasoning failures reflect irreversible loss or arise from recoverable trajectory choices. To this end, we analyze alternative continuations from the same intermediate state around each failure onset.

For a reasoning trajectory τ with a failure onset at segment s_k, we treat the last valid segment, s_{k-1}, as an anchor and generate multiple alternative continuations from the corresponding prefix via stochastic sampling. A failed trajectory is considered *locally recoverable* if at least one alternative continuation from this prefix reaches a correct final answer. This definition focuses on variability in continuation from the same valid prefix, without introducing additional information.

Figure 5 shows that more than 20% trajectories satisfy this criterion. In these cases, correct solutions remain reachable from the same prefix despite failure in the original trajectory, indicating that the error arises from the specific continuation taken after the onset rather than from an absence of viable reasoning paths. Recoverable cases therefore constitute a substantial subset of failures rather than isolated exceptions.

These observations indicate that early failures do not uniquely determine reasoning outcomes. Even when a trajectory diverges and subsequently expands through erroneous reasoning, alternative continuations from the same prefix can still reach correct solutions, highlighting the role of trajectory choice in shaping reasoning behavior.

4 Guided Uncertainty-Aware Inference Control

Motivated by the observed failure dynamics, we propose **G**uided **U**ncertainty-**A**ware **R**easoning with **D**ecision control (**GUARD**), a lightweight test-time approach for intervening during LLM reasoning. GUARD monitors uncertainty signals computed from the model's next-token distribution and triggers intervention only at moments indicative of imminent failure. When triggered, it performs short-horizon branching to obtain a small set of candidate continuations and then selects a continuation based on entropy reduction, avoiding extensive search. In addition, GUARD incorporates a lightweight control mechanism for late-stage reasoning, where prolonged trajectory expansion is unlikely to yield correction. The remainder of this section describes the uncertainty signals used for triggering, the branch-and-select procedure, and the late-stage control mechanism.

4.1 Detecting Failure Onsets

Elevated entropy often coincides with critical transitions that precede reasoning errors, making uncertainty a useful signal for selective intervention. We therefore monitor the token-wise Shannon entropy $\mathcal{H}(x_t)$ during generation and detect atypical spikes relative to the uncertainty observed so far.

To avoid brittle absolute thresholds, we compare the instantaneous entropy to an instance-adaptive baseline defined by a quantile of the entropy history $\mathbf{H}_{<t}$. This relative criterion identifies sharp uncertainty increases under the current prefix while remaining insensitive to the overall entropy scale.

Intervention is evaluated only at reasoning-step boundaries, where a new segment begins and local modifications can be applied without interrupting an ongoing step. Let $\mathcal{T}_{\text{delim}}$ denote the set of delimiter tokens (e.g., \n\n). For the token x_t immediately following such a delimiter, we define

$$\mathbb{I}_{\text{drift}}(x_t) = \mathbb{I}\big[x_{t-1} \in \mathcal{T}_{\text{delim}} \wedge \mathcal{H}(x_t) > \text{Quantile}_q(\mathbf{H}_{<t})\big], \tag{3}$$

where $q \in (0,1)$ controls the sensitivity of the detector. When $\mathbb{I}_{\text{drift}}(x_t) = 1$, GUARD activates the short-horizon branching procedure described in Section 4.2. This detection mechanism restricts intervention to a small number of high-risk transitions, avoiding unnecessary interference during routine generation.

4.2 Branching at Failure Onsets

After a high-uncertainty transition is detected, the goal is to probe a small set of immediate alternatives from the same reasoning state, rather than to diversify generation globally. We therefore apply a localized branching procedure that operates directly on the current prefix.

When the uncertainty trigger is activated ($\mathbb{I}_{\text{drift}}(x_t) = 1$), GUARD performs short-horizon semantic branching from the fixed prefix $x_{<t}$. A small number of candidate continuations are generated in parallel, each limited to a short horizon L. Since all branches share the same prefix, they reuse the pre-computed Key–Value cache of $x_{<t}$, enabling efficient batched generation with minimal latency overhead and only a marginal increase in memory usage. The purpose of branching is to explore distinct local continuations of the same reasoning state, not to approximate an extensive search over solution paths.

We instantiate three complementary branches. (1) **Momentum branch**: Generation proceeds from $x_{<t}$ using standard greedy decoding, preserving the model's current continuation as a reference. (2) **Inhibitory branch**: The token sequence `"Wait,"` is prepended before generation, introducing a brief interruption that disrupts immediate continuation patterns. (3) **Counterfactual branch**: The token sequence `"Let me reconsider:"` is prepended before generation, encouraging a reframing of the next reasoning step while retaining the same prefix.

For each branch, we generate a continuation $c^{(i)}$ and evaluate its uncertainty over the generated horizon. This is summarized by the mean token-level entropy

$$\bar{\mathcal{H}}\left(c_t^{(i)}\right) = \frac{1}{L}\sum_{j=0}^{L-1} \mathcal{H}\left(x_{t+j}^{(i)} \mid x_{<t+j}^{(i)}\right). \quad (4)$$

GUARD selects the continuation with the lowest average entropy,

$$c_t^* = \arg\min_i \bar{\mathcal{H}}\left(c_t^{(i)}\right), \quad (5)$$

and discards the remaining branches. Generation then resumes exclusively from c_t^*.

This branch-and-select procedure is deliberately constrained. By confining branching to a short horizon and collapsing back to a single continuation immediately after selection, this procedure probes local alternatives without maintaining parallel trajectories beyond the intervention window.

4.3 Controlling Late-Stage Reasoning

Incorrect reasoning trajectories often continue to expand in later stages, whereas correct solutions are typically concise. Once a trajectory has entered a prolonged generation phase, further deliberation is unlikely to reverse an earlier error and instead tends to extend unproductive reasoning. We therefore introduce a lightweight mechanism to control late-stage reasoning and favor timely convergence, at which point the branching mechanism introduced earlier is disabled to prevent further expansion.

We characterize the progression of inference using the remaining capacity ratio,

$$\rho_t = 1 - \frac{B_{\text{used}}(t)}{B_{\max}}. \quad (6)$$

which measures how far generation has advanced relative to the maximum allowed length. Smaller values of ρ_t correspond to later stages of generation. Termination control is considered only when ρ_t falls below a threshold $\rho_{\min}$, indicating entry into the late stage.

Within this regime, GUARD monitors the generation stream for hesitation markers that typically precede renewed deliberation. Let $\hat{x}_t$ denote the token predicted by the model at step t and let $\mathcal{T}_{\text{hes}}$ denote a small set of hesitation tokens (e.g., `"Wait"`). When a hesitation marker is produced in the late stage, GUARD replaces the predicted token with a termination signal,

$$x_t = \begin{cases} \texttt{</think>} & \text{if } \hat{x}_t \in \mathcal{T}_{\text{hes}} \,\wedge\, \rho_t \le \rho_{\min}, \\ \hat{x}_t & \text{otherwise.} \end{cases} \quad (7)$$

This design leverages signals already present in the model's generation behavior to suppress further expansion when additional reasoning is unlikely to be beneficial. Restricting termination control to the late stage preserves flexibility during early reasoning while limiting further expansion once continued deliberation becomes unproductive.

5 Experiments

5.1 Setup

Models and Benchmarks. We evaluate our method across model scales using the distilled **DeepSeek-R1-Distill** family (1.5B/7B) (Guo et al., 2025) and the dense **QwQ-32B** (Qwen Team, 2025).

Experiments are conducted on a diverse benchmark suite spanning four evaluation domains: (1) *Competition Reasoning*: AMC23 (AI-MO, 2024), AIME24/25 (MAA Committees, 2025); (2) *Formal Quantitative Reasoning*: MATH500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022); (3) *Coding: LiveCodeBench* (Jain et al., 2024); (4) *Domain Knowledge*: OlympiadBench (He et al., 2024), GPQA Diamond (Rein et al., 2024).

Evaluation Metrics. Following prior work (Zhang et al., 2025b; Xu et al., 2025b), we report **Pass@1** accuracy and the **Average Output Length** (in tokens). We present the mean and standard deviation ($\mu \pm \sigma$) across three independent runs.

Implementation Details. All experiments are conducted on 6 NVIDIA RTX 4090 GPUs using a temperature of 0.0, top-$p = 0.95$, and a maximum budget of $B_{\max} = 10{,}000$ tokens. For GUARD configurations, we set the entropy quantile $q = 0.9$, the late-stage threshold $\rho_{\min} = 0.2$, and the branch-

Table 1: **Performance Comparison Across Multiple Benchmarks.** We report Pass@1 (%) with the average number of generated tokens shown in parentheses. Standard deviations, when available, are indicated as subscripts. Best and second-best results per benchmark are highlighted with Best and Second Best ; overall best and second-best averages are marked with Best and Second Best .

Method	Competition Reasoning			Quantitative		Code	Domain Knowledge		Avg.
	AIME24	AIME25	AMC23	MATH500	Minerva	LiveCode	Olympiad	GPQA	Pass@1
DeepSeek-R1-Distill-Qwen-1.5B									
BASE	20 (8.9k)	13.3 (8.3k)	57$_{\pm 2.6}$ (5.8k)	78.9$_{\pm 1.3}$ (3.8k)	29.5$_{\pm 0.9}$ (5.2k)	17.8 (6.9k)	39.1$_{\pm 2.7}$ (6.3k)	33.8 (7.3k)	36.2 (6.6k)
s1	20 (8.3k)	16.7 (9.1k)	52.5 (6.5k)	78.1$_{\pm 2.5}$ (5.0k)	32.1$_{\pm 0.9}$ (6.1k)	18.4$_{\pm 0.3}$ (7.4k)	42.1$_{\pm 1.1}$ (7.0k)	44.4 (7.9k)	38.0 (7.2k)
CoD	16.7 (7.7k)	16.7 (8.5k)	55.8$_{\pm 2.9}$ (5.6k)	80.2$_{\pm 0.7}$ (3.3k)	30.4$_{\pm 0.5}$ (4.5k)	19.5$_{\pm 0.9}$ (7.2k)	41.4$_{\pm 2.2}$ (6.0k)	45.5 (6.2k)	38.3 (6.1k)
α1	20 (6.8k)	**26.7** (6.8k)	70 (4.3k)	80.4 (3.5k)	31.2 (4.5k)	21.4$_{\pm 0.6}$ (4.8k)	44.2$_{\pm 0.4}$ (5.0k)	35.9 (4.3k)	41.2 (5.1k)
Reflexion	30 (12.8k)	23.3 (12.3k)	**72.5** (8.1k)	80.2$_{\pm 1.0}$ (4.9k)	33.1 (6.9k)	19.3$_{\pm 1.0}$ (15.0k)	**45.5** (9.7k)	46.1 (8.3k)	43.8 (9.8k)
ToT	25.5$_{\pm 13.5}$ (18.0k)	17.8$_{\pm 1.9}$ (17.9k)	58.3$_{\pm 7.2}$ (14.6k)	74.7$_{\pm 2.3}$ (12.1k)	23$_{\pm 2.3}$ (12.8k)	**22.8**$_{\pm 3.3}$ (21.9k)	38.3$_{\pm 2.8}$ (15.1k)	25.8$_{\pm 14.0}$ (12.0k)	35.8 (15.5k)
Best-of-N	30 (35.0k)	20 (34.4k)	67.5 (23.5k)	**81.6** (16.1k)	33.1 (22.3k)	21 (7.7k)	40.7 (27.4k)	**47** (28.7k)	42.6 (24.4k)
Entro-duction	16.7 (6.0k)	16.7 (4.5k)	35.8$_{\pm 10.1}$ (5.4k)	52.2$_{\pm 0.4}$ (3.3k)	13.1$_{\pm 1.7}$ (4.4k)	18.7$_{\pm 0.6}$ (7.1k)	20.2$_{\pm 0.4}$ (4.2k)	40.4 (5.7k)	26.7 (5.1k)
EAGER	**33.3** (16.9k)	23.3 (15.5k)	62.5 (11.2k)	68.6 (6.6k)	15.4 (8.8k)	17.3$_{\pm 0.6}$ (17.5k)	30.8$_{\pm 1.5}$ (12.8k)	41.8 (16.1k)	36.6 (13.2k)
DTS	26.6 (16.8k)	**26.7** (17.0k)	70 (10.7k)	58.1$_{\pm 2.2}$ (6.8k)	22.3$_{\pm 0.2}$ (9.4k)	17.8 (16.6k)	30.1 (13.6k)	40.6$_{\pm 1.0}$ (14.9k)	36.5 (13.2k)
GUARD	**33.3** (9.4k)	**26.7** (8.5k)	**72.5** (6.5k)	81.2$_{\pm 1.0}$ (4.8k)	**34.6** (6.4k)	20.7$_{\pm 1.0}$ (7.7k)	43.7 (7.6k)	**47** (7.6k)	**45.0** (7.3k)
DeepSeek-R1-Distill-Qwen-7B									
BASE	33.3 (8.4k)	26.7 (8.0k)	82.5 (4.7k)	87.5$_{\pm 0.1}$ (3.4k)	39.7 (4.5k)	43.5 (6.0k)	52.6 (6.0k)	44.4 (6.6k)	51.3 (6.0k)
s1	46.7 (8.4k)	26.7 (8.5k)	80 (5.7k)	91 (5.6k)	39.7 (5.3k)	44 (6.7k)	54.2 (6.7k)	43.9 (7.8k)	53.3 (6.8k)
CoD	43.3 (7.5k)	26.7 (7.5k)	85 (3.8k)	91 (2.2k)	40.4 (2.2k)	48.3 (6.4k)	53.8 (5.0k)	45 (5.3k)	54.2 (4.9k)
α1	46.7 (6.8k)	33.3 (6.9k)	82.5 (4.4k)	90 (3.9k)	39.7 (4.3k)	48.3 (5.2k)	**57.5** (5.0k)	47 (4.9k)	55.6 (5.2k)
Reflexion	52.2$_{\pm 3.9}$ (11.9k)	**36.7**$_{\pm 5.8}$ (12.0k)	**90.0** (5.9k)	**92.6** (5.8k)	**42.3** (5.8k)	48.4$_{\pm 0.1}$ (11.1k)	**57.5** (8.3k)	46.1$_{\pm 2.3}$ (7.9k)	58.2 (8.6k)
ToT	47.8$_{\pm 3.9}$ (17.2k)	33.4$_{\pm 5.8}$ (17.5k)	78.3$_{\pm 2.9}$ (13.4k)	87.1$_{\pm 0.1}$ (11.5k)	32.5$_{\pm 1.1}$ (12.0k)	**53.8**$_{\pm 0.4}$ (19.2k)	51.5$_{\pm 0.7}$ (14.6k)	51.7$_{\pm 0.3}$ (13.8k)	54.5 (14.9k)
Best-of-N	36.7 (31.5k)	30 (32.7k)	77.5 (20.0k)	91.2 (13.6k)	41.2 (17.9k)	48 (6.7k)	55.9 (24.1k)	47 (28.7k)	53.4 (21.8k)
Entro-duction	15.6$_{\pm 2.0}$ (6.0k)	16.7 (4.5k)	35.8$_{\pm 10.1}$ (5.4k)	52.2$_{\pm 0.4}$ (3.3k)	13.1$_{\pm 1.7}$ (4.4k)	18.7$_{\pm 0.6}$ (7.1k)	20.2$_{\pm 0.4}$ (4.2k)	40.4 (5.7k)	26.6 (5.1k)
EAGER	**60** (10.9k)	**36.7** (13.4k)	**90** (8.0k)	70.6 (5.3k)	25.7 (5.7k)	47.2$_{\pm 0.3}$ (17.4k)	53 (13.3k)	46 (13.2k)	53.7 (10.9k)
DTS	43.3 (13.7k)	26.7 (15.2k)	**90** (9.7k)	64.8 (4.9k)	33.8 (6.9k)	46.6$_{\pm 1.0}$ (12.8k)	36.6 (11.1k)	46.4 (11.3k)	48.5 (10.7k)
GUARD	**60** (8.5k)	**36.7** (9.2k)	87.5 (5.8k)	90.6 (4.1k)	41.9 (5.3k)	50 (6.6k)	55.9 (6.8k)	**56.6** (7.4k)	**59.8** (6.7k)
Qwen QwQ-32B									
BASE	53.3 (8.7k)	36.7 (8.7k)	77.5 (6.3k)	92.4 (4.0k)	46 (5.2k)	73.8 (6.6k)	58.8 (6.8k)	56.1 (6.7k)	61.8 (6.6k)
s1	46.7 (8.9k)	43.3 (9.0k)	82.5 (6.6k)	91 (4.8k)	48.9 (5.7k)	72 (8.0k)	53.4 (7.7k)	43.9 (7.8k)	60.2 (7.3k)
CoD	63.3 (7.7k)	46.7 (5.3k)	85 (5.1k)	91 (2.8k)	47.4 (3.3k)	76.5 (5.3k)	59.9 (5.7k)	56.1 (5.1k)	65.7 (5.0k)
α1	53.3 (5.7k)	33.3 (6.3k)	87.5 (4.3k)	88.2 (3.2k)	46 (2.9k)	78.25 (6.2k)	54.1 (4.5k)	50.5 (3.5k)	61.4 (4.6k)
Reflexion	63.3 (12.8k)	50 (14.9k)	87.5 (8.2k)	**95.2** (4.7k)	50 (7.3k)	79.75 (7.6k)	66.2 (10.3k)	**59.2** (8.5k)	68.9 (9.3k)
ToT	47.8$_{\pm 3.9}$ (17.9k)	31.1$_{\pm 5.1}$ (18.0k)	85 (15.7k)	92 (13.3k)	44.9 (14.3k)	**82.4**$_{\pm 6.7}$ (19.7k)	59.3 (16.2k)	58.9$_{\pm 3.7}$ (15.0k)	62.7 (16.3k)
Best-of-N	55.6$_{\pm 2.0}$ (32.4k)	36.7 (32.4k)	78.5 (19.9k)	92.4 (13.4k)	47.4 (17.9k)	76.5 (6.8k)	59.9 (24.0k)	57.6$_{\pm 3.5}$ (27.8k)	63.1 (21.8k)
Entro-duction	15.6$_{\pm 2.0}$ (6.0k)	16.7 (4.5k)	35.8$_{\pm 10.1}$ (5.4k)	52.2$_{\pm 0.4}$ (3.3k)	13.1$_{\pm 1.7}$ (4.4k)	18.7$_{\pm 0.6}$ (7.1k)	20.2$_{\pm 0.4}$ (4.2k)	40.4 (5.7k)	26.6 (5.1k)
EAGER	47.8$_{\pm 1.9}$ (9.3k)	36.7 (9.8k)	77.5 (6.9k)	71.8 (4.6k)	25.7 (6.5k)	76.5 (7.7k)	56.0$_{\pm 3.4}$ (8.0k)	46 (13.2k)	54.8 (8.2k)
DTS	63.3 (14.8k)	46.7 (15.0k)	**92.5** (10.4k)	84.8 (5.3k)	37.9 (6.9k)	77.75 (12.2k)	58.8$_{\pm 1.9}$ (12.2k)	53.3 (11.4k)	64.4 (11.0k)
GUARD	**76.7** (9.2k)	**53.3** (9.4k)	**92.5** (7.2k)	93 (4.9k)	**50.4** (6.5k)	80 (6.5k)	**69.8** (8.9k)	54.5 (7.5k)	**71.3** (7.5k)
Transferability on Math-Specialized Model									
JustRL	40 (7.4k)	24.4$_{\pm 2.0}$ (7.2k)	77.5 (5.4k)	87.4 (4.0k)	35.7 (5.1k)	17 (7.3k)	51 (6.0k)	29.8 (5.0k)	45.4 (5.9k)
+ GUARD	**46.7** (7.8k)	**30** (8.0k)	**87.5** (5.7k)	87.4 (5.0k)	**38.6** (6.9k)	**32** (7.9k)	**52.9** (7.3k)	**34.8** (7.9k)	**51.2** (7.1k)

ing horizon $L = 200$ tokens. The hesitation trigger is set to $\mathcal{T}_{\text{hes}} = \{\texttt{"Wait"}\}$, while $\mathcal{T}_{\text{delim}}$ targets structural boundaries (e.g., "\n\n"; full list in Appendix B).

5.2 Main Results

Table 1 evaluates GUARD on reasoning-oriented models, comparing it with single-trajectory optimization methods (CoD, s1, α1, Reflexion), and parallel search paradigms (Best-of-N, ToT, Entro-duction, EAGer, DTS). Detailed configurations are provided in Appendix B.4. Across all model scales (1.5B, 7B, and 32B), GUARD consistently achieves the strongest accuracy–length trade-off. In particular, on the 32B model, GUARD attains 71.3% Pass@1 using only ∼7.5k generated tokens, indicating that strong reasoning performance does not require exhaustive parallel sampling or repeated full-chain regeneration.

These gains stem from GUARD's selective, lightweight intervention. Unlike Reflexion, which relies on external correctness signals and repeated reprocessing that incurs additional inference latency, and parallel search methods, which expand

Table 2: **Performance on a General Instruction-Tuned Backbone** Results on Llama-3.1-8B-Instruct compare GUARD with Self-Consistency (Wang et al., 2023), SELF-REFINE (Madaan et al., 2023), and EM-INF (Agarwal et al., 2025), highlighting effectiveness beyond reasoning-specialized models. Baseline details are in Appendix B.4.3.

Method	Math	AMC	AIME	Minerva	Olymp.	Avg.
Llama-3.1-8B-Instruct	40.6	18.1	1.1	22.4	15.7	19.6
Greedy Decoding	40.6	16.9	3.3	21.0	16.0	19.6
SELF-REFINE	41.0	19.3	1.1	22.4	15.7	19.9
Self-consistency	41.2	20.5	4.4	20.2	19.4	21.1
Adaptive Temp	43.6	25.3	5.5	**24.3**	16.6	23.1
EM-INF	43.0	22.9	3.3	22.8	16.4	21.7
GUARD	**49.5**	**32.5**	**6.7**	23.2	**21.7**	**26.7**

Table 3: **Ablation Analysis of GUARD.** We report average Pass@1 accuracy across eight benchmarks using DeepSeek-R1-Distill-7B. The Δ column shows the absolute performance change relative to the full GUARD configuration.

Configuration	Acc. (%)	Δ
Full GUARD	**59.8**	-
Internal Components		
w/o Counterfactual	55.4	-4.4
w/o Inhibitory	57.1	-2.7
w/o Momentum	54.3	-5.5
Late-stage Reasoning Control		
w/o Late-stage Control	53.0	-6.8

computation and disrupt long-range coherence, GUARD intervenes only at high-risk transitions. By using adaptive, instance-specific uncertainty signals to trigger short-horizon branching, GUARD corrects trajectories efficiently without maintaining parallel paths or relying on fixed thresholds.

5.3 Transferability Across Backbone Types

We further evaluate the generality of GUARD beyond the reasoning-oriented backbones used in our main experiments. Specifically, we consider two complementary settings: (1) domain-specialized backbones extensively fine-tuned for a specific task, and (2) general-purpose instruction-tuned backbones without explicit reasoning optimization. These experiments assess whether GUARD functions as a plug-in inference-time mechanism independent of the backbone's training paradigm.

Math-Specialized Backbones. We apply GUARD to JustRL-1.5B (He et al., 2025), a math-specialized model fine-tuned from *DeepSeek-R1-Distill-1.5B*. While such specialization yields strong in-domain performance, it often reduces robustness on non-math tasks. As shown in Table 1, GUARD consistently improves mathematical accuracy while also recovering performance on out-of-domain tasks such as coding, indicating that it complements domain-specific fine-tuning through inference-time correction.

General Instruction-Tuned Backbones. We further evaluate GUARD on Llama-3.1-8B-Instruct and compare it with EM-INF (Agarwal et al., 2025), an unsupervised method that reduces entropy globally. In contrast to this global strategy, GUARD intervenes selectively at high-risk transitions. As reported in Table 2, GUARD consistently outperforms EM-INF and other baselines (Appendix B.4.3), demonstrating effectiveness even when the backbone is not optimized for structured reasoning.

5.4 Ablation Analysis

Table 3 reports ablation results on DeepSeek-R1-Distill-7B, averaged across eight benchmarks, with full configurations and per-benchmark results provided in Appendix C.

Component Contribution. All three branching components are necessary for strong performance. Removing any of the Momentum, Inhibitory, or Counterfactual branches consistently degrades accuracy, indicating that effective rectification depends on their complementary roles.

Role of Late-Stage Control. Late-stage control is critical for preventing performance collapse. Disabling this mechanism leads to prolonged deliberation near the end of generation and a marked drop in performance.

Hyperparameter Sensitivity. GUARD exhibits stable performance across a wide range of hyperparameter settings. Sensitivity analyses for entropy quantiles, branching horizons, and termination thresholds are reported in Appendix D.

Conclusion

Reasoning performance often improves with increased inference-time computation, yet failure dynamics remain underexplored. We show that errors originate from a few early transitions marked by entropy spikes and propagate through coherent reasoning, motivating GUARD, a targeted inference-time framework that intervenes at high-risk transitions via brief local branching. Our results highlight that identifying where reasoning first deviates complements scaling-based approaches.

Limitations

Our analysis focuses on trajectory-level failure dynamics under a controlled setup. Segment validity relies on an external oracle and token-level entropy is used as the primary uncertainty signal, which may not capture all forms of reasoning difficulty. Experiments emphasize structured reasoning benchmarks, and how these patterns extend to more open-ended domains or training-time integration remains to be explored.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. 2025. The unreasonable effectiveness of entropy minimization in llm reasoning. *arXiv preprint arXiv:2505.15134*.

AI-MO. 2024. AIMO Validation Dataset - AMC. https://huggingface.co/datasets/AI-MO/aimo-validation-amc. Accessed: 2024-11-19.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 17682–17690.

Zhuokun Chen, Zeren Chen, Jiahao He, Lu Sheng, Mingkui Tan, Jianfei Cai, and Bohan Zhuang. 2025. R-stitch: Dynamic trajectory stitching for efficient reasoning. *arXiv preprint arXiv:2507.17307*.

Tianyu Fu, Yi Ge, Yichen You, Enshu Liu, Zhihang Yuan, Guohao Dai, Shengen Yan, Huazhong Yang, and Yu Wang. 2025. R2r: Efficiently navigating divergent reasoning paths with small-large model token routing. *arXiv preprint arXiv:2505.21600*.

Google DeepMind. 2025. A new era of intelligence with gemini 3. https://blog.google/products/gemini/gemini-3/. Accessed: 2025-11-29.

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*.

Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, and 1 others. 2025. Justrl: Scaling a 1.5 b llm with a simple rl recipe. *arXiv preprint arXiv:2512.16649*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850.

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*.

Jiameng Huang, Baijiong Lin, Guhao Feng, Jierun Chen, Di He, and Lu Hou. 2025. Efficient reasoning for large reasoning language models via certainty-guided reflection suppression. *arXiv preprint arXiv:2508.05337*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857.

Xianzhi Li, Ethan Callanan, Abdellah Ghassel, and Xiaodan Zhu. 2025. Entropy-gated branching for efficient test-time reasoning. *arXiv preprint arXiv:2503.21961*.

Baohao Liao, Yuhui Xu, Hanze Dong, Junnan Li, Christof Monz, Silvio Savarese, Doyen Sahoo, and Caiming Xiong. 2025. Reward-guided speculative decoding for efficient llm reasoning. *arXiv preprint arXiv:2501.19324*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

MAA Committees. 2025. AIME Problems and Solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions. Accessed: 2025-11-19.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback, 2023. *URL https://arxiv.org/abs/2303.17651*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candes, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20286–20332, Suzhou, China. Association for Computational Linguistics.

Qwen Team. 2025. QwQ-32B-Preview: Preview of Qwen QwQ-32B. `https://qwenlm.github.io/blog/qwq-32b-preview/`. Accessed: 2025-03-20.

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*.

Daniel Scalena, Leonidas Zotos, Elisabetta Fersini, Malvina Nissim, and Ahmet Üstün. 2025. Eager: Entropy-aware generation for adaptive inference-time scaling. *arXiv preprint arXiv:2510.11170*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*.

Qwen Team and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3).

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yi Wang, Junxiao Liu, Shimao Zhang, Jiajun Chen, and Shujian Huang. 2025. Pats: Process-level adaptive thinking mode switching. *arXiv preprint arXiv:2505.19250*.

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.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025a. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*.

Zicheng Xu, Guanchu Wang, Yu-Neng Chuang, Guangyao Zheng, Alexander S Szalay, Zirui Liu, and Vladimir Braverman. 2025b. Dts: Enhancing large reasoning models via decoding tree sketching. *arXiv preprint arXiv:2511.00640*.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. 2025a. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*.

Wang Yang, Xiang Yue, Vipin Chaudhary, and Xiaotian Han. 2025b. Speculative thinking: Enhancing small-model reasoning with large model guidance at inference time. *arXiv preprint arXiv:2504.12329*.

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.

Xixian Yong, Xiao Zhou, Yingying Zhang, Jinlin Li, Yefeng Zheng, and Xian Wu. 2025. Think or not? exploring thinking efficiency in large reasoning models via an information-theoretic lens. *arXiv preprint arXiv:2505.18237*.

Jinghan Zhang, Xiting Wang, Fengran Mo, Yeyang Zhou, Wanfu Gao, and Kunpeng Liu. 2025a. Entropy-based exploration conduction for multi-step reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3895–3906, Vienna, Austria. Association for Computational Linguistics.

Junyu Zhang, Runpei Dong, Han Wang, Xuying Ning, Haoran Geng, Peihao Li, Xialin He, Yutong Bai, Jitendra Malik, Saurabh Gupta, and Huan Zhang. 2025b. AlphaOne: Reasoning models thinking slow and fast at test time. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 11340–11365, Suzhou, China. Association for Computational Linguistics.

Wei Zhu, Zhiwen Tang, and Kun Yue. 2025. Symphony: Synergistic multi-agent planning with heterogeneous language model assembly. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

APPENDIX

A GUARD Inference Algorithm

Algorithm 1 outlines the complete execution workflow of the GUARD framework.

B Detailed Experimental Setup

This appendix provides a comprehensive description of the evaluation metrics, benchmarks, and baseline methods used in our experiments.

B.1 Evaluation Metrics

We employ two primary metrics to assess reasoning performance and computational efficiency:

Pass@1 Accuracy. To facilitate consistent evaluation across all models and benchmarks, we explicitly instruct models via the system prompt to enclose their final answer within `\boxed{}`. We extract the content inside these tags for verification. For open-ended quantitative tasks (e.g., MATH, AIME), we compare the extracted value against the ground truth using symbolic equivalence checks (e.g., `sympy`) to account for notational invariance. For multiple-choice tasks (e.g., GPQA), we perform exact string matching on the extracted option key. An output is deemed correct only if the boxed content strictly matches the ground truth label.

Average Output Length. To quantify inference efficiency, we measure the total number of tokens generated per query. This includes the entire chain-of-thought reasoning trace and the final answer, but excludes the input prompt tokens. Lower token consumption indicates higher efficiency. For methods involving parallel sampling or tree search, the token count is the sum of tokens generated across all sampled paths or tree branches for a single query.

B.2 GUARD Implementation Details

In addition to the decoding parameters specified in the main text, we provide the precise definitions of the token sets used for failure detection and intervention triggering.

Hyperparameters. The specific thresholds used for the GUARD controller are: entropy quantile sensitivity $q = 0.9$, late-stage budget threshold $\rho_{\min} = 0.2$, and a short-horizon branching limit of $L = 200$ tokens.

Token Definitions. The detection mechanism relies on two specific sets of tokens. Note that we represent the newline character as `\n`.

- **Hesitation Set ($\mathcal{T}_{\text{hes}}$).** This set targets explicit linguistic markers of stalling or hesitation generated by the model:

$$\mathcal{T}_{\text{hes}} = \{\texttt{"Wait"}\}$$

- **Delimiter Set ($\mathcal{T}_{\text{delim}}$).** This set identifies structural boundaries (e.g., end of paragraphs or logic blocks) where interventions are permitted. It includes standard double-newlines and their combinations with punctuation:

$$\mathcal{T}_{\text{delim}} = \begin{Bmatrix} \texttt{"\textbackslash n\textbackslash n"}, & \texttt{",\textbackslash n\textbackslash n"}, & \texttt{".\textbackslash n\textbackslash n"}, \\ \texttt{"]\textbackslash n\textbackslash n"}, & \texttt{")\textbackslash n\textbackslash n"}, & \texttt{"]),\textbackslash n\textbackslash n"}, \\ \texttt{"].\textbackslash n\textbackslash n"}, & \texttt{").\textbackslash n\textbackslash n"}, & \texttt{".)\textbackslash n\textbackslash n"} \end{Bmatrix}$$

B.3 Benchmark Details

Our evaluation suite encompasses four cognitive domains, utilizing datasets specifically chosen for their rigor and ability to differentiate high-capability reasoning models.

Competition Reasoning. This category evaluates the model's ability to navigate complex, non-routine problems requiring creative heuristics and multi-step planning.

- **AMC 2023** (AI-MO, 2024): A dataset consisting of 40 problems selected from the 2023 AMC 12A and 12B contests. Sponsored by the Mathematical Association of America, these exams target U.S. students in grade 12 and below, featuring challenges across algebra, geometry, number theory, and combinatorics.

- **AIME 2024 & 2025** (MAA Committees, 2025): A specialized benchmark collection consisting of 60 problems in total—30 from the 2024 American Invitational Mathematics Examination (AIME) and 30 from the 2025 edition. These problems cover core secondary-school mathematics topics but place rigorous demands on both solution accuracy and conceptual depth, serving as a robust test for advanced mathematical reasoning.

Formal Quantitative. These benchmarks assess the model's command over standard academic axioms and symbolic manipulation.

- **MATH500** (Hendrycks et al., 2021): A curated selection of 500 problems extracted from the MATH benchmark. The collection covers a wide range of high-school mathematics domains, including Prealgebra, Algebra, and Number Theory. To ensure comparability with prior work, we utilize the exact problem set originally curated by OpenAI for evaluation.

- **Minerva Math** (Lewkowycz et al., 2022): This dataset consists of 272 undergraduate-level STEM problems harvested from MIT's OpenCourseWare, specifically designed to evaluate multi-step scientific reasoning. The problems span solid-state chemistry, information and entropy, differential equations, and special relativity. Each problem includes a clearly delineated answer—191 verifiable by numeric checks and 81 by symbolic solutions.

Coding.

- **LiveCodeBench** (Jain et al., 2024): A contamination-free benchmark for evaluating large language models on code generation. The suite is continuously updated to mitigate data leakage. For this study, we utilize the subset comprising 400 Python programming tasks released between May 2023 and March 2024. Each task is paired with test samples for correctness verification. Beyond basic generation, this benchmark implicitly measures advanced capabilities such as self-repair and edge-case handling.

Domain Knowledge. This category tests the model's ability to synthesize expert-level knowledge across interdisciplinary fields.

- **OlympiadBench** (He et al., 2024): A comprehensive dataset evaluating mathematical and physical reasoning at the Olympiad level. It features a wide difficulty range and expert solution annotations. From the original 8,476 problems, we utilize a specific subset of 675 open-ended, text-only math competition problems in English to focus on pure reasoning without multimodal dependencies.

- **GPQA Diamond** (Rein et al., 2024): A PhD-level benchmark consisting of high-quality questions spanning physics, chemistry, and biology subdomains. The dataset is notably difficult; domain experts with PhDs in these respective fields achieved only 69.7% accuracy. We specifically select the highest-quality subset, GPQA Diamond (198 questions), to strictly evaluate the model's capacity for expert-level scientific reasoning and knowledge retrieval.

Algorithm 1 GUARD Inference Process

Require: Model $\mathcal{M}$, Prompt $x_{<1}$, Budget $B_{\max}$, Horizon L

Initialize $t \leftarrow 0$, sequence $x \leftarrow x_{<1}$, entropy history $\mathbf{H} \leftarrow \emptyset$
while $t < B_{\max}$ **and not** EOS **do**
  Sample candidate $\hat{x}_t$ and compute entropy h_t from $\mathcal{M}(x)$
  Update budget ratio $\rho = 1 - t/B_{\max}$
  // 1. Late-Stage Control (Sec. 4.3)
  if $\rho \leq \rho_{\min}$ **and** $\hat{x}_t \in \mathcal{T}_{\text{hes}}$ **then**
    $x \leftarrow x +$ </think> ▷ Force termination
    continue
  end if
  // 2. Failure Detection (Sec. 4.1)
  Let x_{last} be the last token of x
  if $x_{last} \in \mathcal{T}_{\text{delim}}$ **and** $h_t >$ Quantile$_q(\mathbf{H})$ **then**
    // 3. Branch-and-Select (Sec. 4.2)
    Generate 3 branches $\{c^{(i)}\}_{i=1}^{3}$ of length L:
      • **Momentum**: Greedy from x
      • **Inhibitory**: Prepend `"Wait,"`
      • **Counterfactual**: Prepend `"Let me.."`
    Select $c^* = \arg\min_i \bar{\mathcal{H}}(c^{(i)})$ ▷ Min Entropy
    $x \leftarrow x + c^*$
    $t \leftarrow t + L$
  end if
  // Standard Generation
  $x \leftarrow x + \hat{x}_t$
  Append h_t to $\mathbf{H}$
  $t \leftarrow t + 1$
end while
return x

B.4 Baseline Descriptions

We compare GUARD against a wide range of inference-time optimization strategies, categorized into single-stream optimizations and parallel search paradigms.

B.4.1 Single-Stream Optimizations

These methods aim to improve reasoning within a single decoding trajectory without maintaining multiple active hypotheses.

- *CoD (Chain of Draft)* (Xu et al., 2025a): A prompting strategy that instructs the model to generate a concise "draft" plan before executing the full reasoning chain. This separates planning from execution to reduce logic errors.

- *s1* (Muennighoff et al., 2025): A budget-forcing method that artificially induces longer deliberation by appending specific wait markers (e.g., "Wait,") to the generation stream. To ensure a fair comparison with other inference-time interventions (following the protocol of α1), we apply s1 directly at test-time as a budget-forcing mechanism *without* the supervised fine-tuning (SFT) stage typically associated with its original implementation.

- *α1* (Zhang et al., 2025b): A framework that modulates reasoning duration via a hyperparameter α. It treats the insertion of transition tokens as a stochastic process before the α moment, after which it forces deterministic termination of the thought process. We use fixed α values tuned specifically for each benchmark.

- *Reflexion* (Shinn et al., 2023): An iterative self-correction framework where the model critiques and modifies its own output. In our experiments, we employ an oracle-based trigger: reflection is initiated only when the generated answer does not match the ground truth. To strictly align with our evaluation metric of generative token consumption, we report the cumulative sum of output tokens produced across all iteration steps. Crucially, we exclude all prompt tokens (including re-ingested error trajectories and reflection instructions) from this calculation to focus solely on the generative cost.

B.4.2 Parallel Search Paradigms

These methods leverage computational redundancy to explore a broader solution space.

- *Best-of-N (BoN)* (Wang et al., 2023): Adopting the standard self-consistency mechanism, we generate $N = 4$ complete independent reasoning paths in parallel for each query. Unlike tree-based methods that evaluate intermediate steps, this approach produces full trajectories before assessment. The final answer is determined via majority voting over the answers extracted from these four parallel candidates.

- *ToT (Tree of Thoughts)* (Yao et al., 2023): A structured search algorithm that explores the reasoning space by decomposing problems into intermediate steps. We implement ToT using a Depth-First Search (DFS) strategy, where the LLM itself serves as the value function to assign quantitative scores to each intermediate node, guiding the pruning and expansion process. Consistent with the Reflexion baseline, our cost metric accounts solely for the cumulative generated tokens across all visited branches, strictly excluding prompt tokens used for state representation and scoring instructions.

- *Entro-duction* (Zhang et al., 2025a): A dynamic framework that adjusts reasoning exploration depth by monitoring two uncertainty metrics: the model's output entropy (current step uncertainty) and variance entropy (fluctuation across steps). Based on these signals, the method probabilistically determines whether to deepen the current reasoning path, expand the search space, or terminate exploration. For our implementation, we adhere to the recommended settings with a maximum depth of 20 steps, an exploration rate of 0.25, and a soft-stop buffer of 2 steps.

- *EAGER* (Scalena et al., 2025): A training-free method that optimizes the efficiency-performance trade-off by dynamically allocating computation based on prompt complexity. Grounded in the assumption that fixed-budget parallel sampling is inefficient for varying problem difficulties, EAGER triggers branching only when detecting high-entropy peaks to concentrate exploration on uncertain steps. While the full framework includes a dataset-level budget reallocation mechanism, we focus on independent per-instance inference. Therefore, we execute only the EAGER-init stage (the preparatory branching phase), using the optimal math configuration: temperature 0.6, entropy threshold 2.2, and a sequence cap of $M = 3$.

- *DTS* (Xu et al., 2025b): A framework that constructs a decoding tree by spawning K parallel branches only when the next-token entropy exceeds a threshold τ. A major limitation of the original study is that its efficacy was validated exclusively on the AIME benchmark using fixed hyperparameters, lacking adaptation guidelines for diverse domains. Consequently, we are constrained to applying their fixed threshold ($\tau = 2.5$) across all our datasets. Additionally, to ensure a fair comparison regarding computational overhead, we restrict the maximum branching factor to $K = 3$.

B.4.3 Baselines for Transferability Analysis

To validate the universality of GUARD across distinct model paradigms (as discussed in Section 5.3), we incorporate two additional baselines representing differing optimization strategies: hyper-specialized RL training and generalist inference-time optimization.

- *JustRL* (He et al., 2025): A minimalist reinforcement learning framework that challenges the necessity of complex multi-stage pipelines. By employing single-stage training with fixed hyperparameters, JustRL achieves state-of-the-art mathematical reasoning performance on 1.5B scale models while using significantly less compute than traditional methods. We utilize the **JustRL-1.5B** checkpoint (derived from *DeepSeek-R1-Distill-1.5B*) to represent *hyper-specialized models*. Our experiment aims to verify whether GUARD can mitigate the "capability tax"—the degradation of out-of-distribution skills (e.g., coding) often induced by such aggressive domain-specific optimization.

- *EM-INF* (Agarwal et al., 2025): A specific inference-time variant within the entropy minimization framework that employs logit adjustment to minimize output entropy without parameter updates. Unlike its fine-tuning counterparts, EM-INF requires no labeled data. We select this method as the baseline for generalist models (specifically Llama-3.1-8B-Instruct) because it represents the state-of-the-art for training-free optimization, offering the fairest comparison to our inference-only approach. The reported results for this baseline are directly referenced from the original publication.

- *Greedy Decoding*: The standard deterministic generation approach where the sampling temperature is strictly fixed at zero. At each step, the model selects the token with the highest probability, establishing a lower-bound baseline for reasoning stability.

- *Self-Consistency* (Wang et al., 2023): A parallel sampling strategy designed to marginalize out reasoning errors. Adhering to the evaluation protocol in (Agarwal et al., 2025), we generate four independent reasoning paths using the prescribed stochastic sampling parameters and derive the final answer via majority voting. This baseline tests whether simple aggregation can outperform targeted steering.

- *SELF-REFINE* (Madaan et al., 2023): A sequential optimization approach where the model's generated output is fed back into the context to prompt self-correction. We implement this feedback loop for three consecutive iterations following the baseline settings in (Agarwal et al., 2025), allowing the model to critique and refine its prior outputs.

- *Adaptive Temperature*: An entropy-aware scaling technique used as a baseline against EM-INF. Instead of using a fixed scalar, this method dynamically reduces the softmax temperature during generation until the output distribution's entropy aligns with the target threshold defined in (Agarwal et al., 2025). This sharpens the distribution without the direct gradient-based logit updates used in EM-INF.

C Additional Ablation Results

In this section, we provide the fine-grained numerical breakdown of the ablation study summarized in Section 5.4. Table 4 details the performance of GUARD on the **DeepSeek-R1-Distill-7B** model across all eight benchmarks when individual branching primitives are removed.

Impact of Branching Primitives. Consistent with the aggregated results in the main text, we observe that removing any single branch type—*Momentum*, *Inhibitory*, or *Counterfactual*—results in performance degradation across the majority of domains. This reinforces our hypothesis that these strategies probe distinct reasoning subspaces:

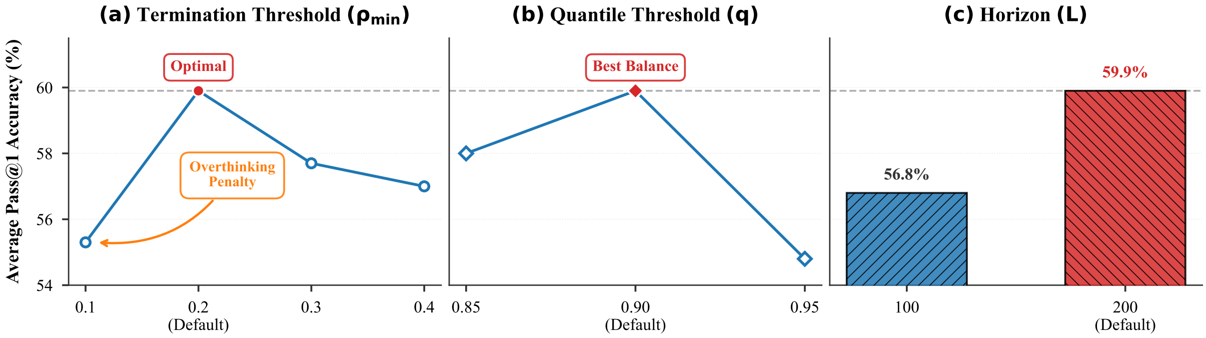

Figure 6: **Analysis of Hyperparameter Choices.** We analyze how key configuration choices influence model performance. **(a) Termination Threshold** $\rho_{\min}$: Performance peaks at $\rho_{\min} = 0.2$. Lower values (0.1) intervene too late, exposing the model to the risk of an "epistemic spiral" in uncontrolled late-stage reasoning. **(b) Quantile Threshold** q: $q = 0.90$ effectively captures failure onsets without excessive triggering. **(c) Horizon** L: A moderate horizon ($L = 200$) is chosen as it is sufficient for accurate branch selection, balancing task performance with computational cost.

Table 4: **Comprehensive Ablation and Sensitivity Analysis.** This table details the performance impact of removing specific branching primitives (Ablation) and varying key hyperparameters (Sensitivity). The *Full Method* (GUARD) adopts the optimal configuration ($L = 200, \rho_{\min} = 0.2, q = 0.90$). Deviating from these settings, such as restricting the horizon, altering termination timing, or changing the quantile threshold, consistently results in suboptimal performance. Results are reported as Pass@1 (%) with average token usage (k), and all figures are rounded to one decimal place. For each benchmark, the best and second-best results are highlighted using Best and Second Best, respectively. In the Average column, the overall best and second-best methods are distinctly marked with Best and Second Best.

Configuration	Competition Reasoning			Quantitative		Code	Domain Knowledge		Avg.
	AIME24	AIME25	AMC23	MATH500	Minerva	LiveCode	Olympiad	GPQA	Pass@1
Ablation: Branching Primitives									
w/o Momentum	46.7 (7.5k)	30.0 (8.2k)	77.5 (5.3k)	86.4 (4.2k)	41.2 (4.9k)	47.2 (5.6k)	50.6 (6.6k)	55.0 (5.2k)	54.3 (5.9k)
w/o Inhibitory	46.7 (7.3k)	**36.7** (7.9k)	80.0 (5.1k)	86.4 (3.9k)	42.6 (4.8k)	48.3 (5.7k)	51.7 (6.2k)	**64.7** (5.2k)	57.1 (5.8k)
w/o Counterfactual	46.7 (7.4k)	30.0 (7.9k)	80.0 (5.2k)	87.2 (4.0k)	40.1 (4.8k)	48.6 (5.6k)	51.3 (6.2k)	59.1 (6.4k)	55.4 (5.9k)
Sensitivity: Branching Horizon (L)									
Horizon $L = 100$	50.0 (8.0k)	**36.7** (8.2k)	80.0 (4.8k)	89.2 (3.1k)	40.4 (4.1k)	48.3 (5.6k)	53.5 (5.9k)	56.6 (6.2k)	56.8 (5.7k)
Sensitivity: Termination Threshold ($\rho_{\min}$)									
Threshold $\rho_{\min} = 0.1$	46.7 (8.5k)	26.7 (9.9k)	82.5 (6.8k)	88.2 (5.1k)	**45.2** (5.7k)	46.1 (7.8k)	53.8 (7.2k)	53.0 (8.2k)	55.3 (7.4k)
Threshold $\rho_{\min} = 0.3$	53.3 (8.3k)	30.0 (9.2k)	**90.0** (5.5k)	88.8 (4.1k)	41.5 (5.1k)	48.3 (6.4k)	54.1 (6.8k)	55.6 (7.1k)	57.7 (6.6k)
Threshold $\rho_{\min} = 0.4$	43.3 (8.7k)	30.0 (8.9k)	**90.0** (4.6k)	90.4 (3.8k)	42.6 (4.7k)	48.3 (6.2k)	55.9 (5.5k)	55.6 (7.1k)	57.0 (6.2k)
w/o Late-Stage Control	36.7 (9.1k)	30.0 (9.9k)	80.0 (7.1k)	86.4 (5.6k)	42.6 (6.8k)	45.5 (8.1k)	51.7 (7.8k)	51.0 (8.9k)	53.0 (7.9k)
Sensitivity: Quantile Threshold (q)									
Quantile $q = 0.85$	53.3 (9.5k)	33.3 (9.4k)	85.0 (6.1k)	**90.6** (7.0k)	40.8 (6.1k)	**50.0** (8.0k)	**56.6** (7.0k)	54.0 (8.5k)	58.0 (7.7k)
Quantile $q = 0.95$	36.7 (7.9k)	30.0 (7.5k)	87.5 (5.3k)	88.0 (4.0k)	41.9 (5.1k)	49.2 (5.3k)	55.6 (6.2k)	49.5 (6.7k)	54.8 (6.0k)
GUARD (Full Method)	**60** (8.5k)	**36.7** (9.2k)	87.5 (5.8k)	**90.6** (4.1k)	41.9 (5.3k)	**50** (6.6k)	55.9 (6.8k)	56.6 (7.4k)	**59.8** (6.7k)

- The **Momentum branch** leverages the model's intrinsic generation inertia to preserve valid partial reasoning.
- The **Inhibitory branch** (induced by "Wait,") effectively disrupts premature convergence and "system 1" thinking patterns.
- The **Counterfactual branch** (induced by "Let me reconsider:") explicitly encourages the model to explore alternative logical paths from the same context.

The results demonstrate that the synergy of these three primitives is essential for maximizing the coverage of the search space, ensuring that valid rectification paths are discovered across diverse failure modes—from symbolic manipulation errors in MATH/Minerva to logic flaws in LiveCodeBench.

D Analysis of Hyperparameter Choices

We characterize the behavior of GUARD under varying configurations to validate our design choices regarding the termination threshold ($\rho_{\min}$), quantile threshold (q), and horizon length (L). As shown in Figure 6, our default configuration represents a balanced operating point that maximizes accuracy while maintaining inference efficiency. Detailed numerical results for all hyperparameter configurations are provided in Table 4.

Termination Threshold ($\rho_{\min}$). Figure 6(a) validates our choice of $\rho_{\min} = 0.2$. The model achieves peak performance (59.9%) when control is activated upon entering the final 20% of the budget. In contrast, delaying intervention ($\rho_{\min} = 0.1$) causes a sharp performance drop to 55.3%. This confirms that unconstrained deliberation in the final stages incurs an "epistemic spiral," necessitating accelerated convergence. Conversely, activating control too early ($\rho_{\min} \geq 0.3$) also reduces accuracy (57.7%) by limiting the reasoning depth required for complex problems.

Quantile Threshold (q) & Horizon (L). Figures 6(b) and (c) justify our selection of branching parameters. A quantile of $q = 0.90$ provides the most effective signal-to-noise ratio for failure detection. Regarding the horizon, we adopt $L = 200$. While significantly longer horizons might offer theoretical benefits, we observe that $L = 200$ is sufficient for reliable entropy estimation. We deliberately chose not to increase L further to preserve the lightweight nature of our inference-time intervention.

E Qualitative Analysis of Epistemic Spirals

We present a qualitative comparison across three model scales (1.5B, 7B, 32B) and domains (Geometry, Physics, Number Theory) to demonstrate GUARD's versatility. We observe that "Epistemic Spirals" manifest differently depending on model capability. As shown below, GUARD identifies these failures via entropy spikes and intervenes with context-aware branching to restore convergence.

1. Overcoming Arithmetic Hesitation (Figure 7). In high-precision geometry (AIME 2024), smaller models like **DeepSeek-R1-Distill-Qwen-1.5B** often falter when facing complex arithmetic. As shown in Figure 7, the model derives the correct equations but enters a loop of self-doubt due to large coefficients ($> 10^7$). GUARD detects this hesitation and injects a Counterfactual Branch, enforcing the execution of the calculation to reveal the integer solution that the base model initially abandoned.

2. Resolving Intuition Conflicts (Figure 8). In physics reasoning (Minerva), even capable models like **DeepSeek-R1-Distill-Qwen-7B** struggle when correct results contradict training priors. Figure 8 illustrates a case where the model doubts a valid but counter-intuitive result (a macroscopic atomic wavelength), triggering unnecessary error checking. GUARD intervenes with a Scaling Law Verification, guiding the model to validate the result via first-principles estimation rather than rejecting the correct path.

3. Shifting form Brute-force to Structure (Figure 9). In number theory (OlympiadBench), larger models like **QwQ-32B** may attempt to solve structural problems through inefficient enumeration. As depicted in Figure 9, the base model wastes tokens searching for non-existent counterexamples. GUARD detects the lack of logical progression and injects a Counterfactual Branch, steering the model away from aimless guessing toward a rigorous proof based on modular arithmetic.

F Use of AI Assistants

We utilized AI assistants to help with language editing and writing refinement. All technical content, experimental results, and scientific claims were verified by the authors.

G Artifacts Statements

G.1 Model Artifacts

We utilize the following models in our work, complying with all respective license terms:

- DeepSeek-R1-Distill-Qwen-1.5B and DeepSeek-R1-Distill-Qwen-7B: Both models are released under the MIT License, which permits commercial use, modification, and redistribution. These models are distilled from the Qwen-2.5 series (Apache 2.0 License).

- Qwen QwQ-32B: This model is released under the Apache License 2.0, allowing both research and commercial usage.

- Llama-3.1-8B-Instruct: Used for our generalist model transferability analysis, this model is released under the Llama 3.1 Community License. We comply with the usage policy and acceptable use guidelines provided by Meta.

G.2 Data Artifacts

We employ publicly available benchmarks for evaluation, including MATH-500, LiveCodeBench, OlympiadBench, GPQA, AIME, AMC 23, and Minerva. These datasets are widely accessible in the open-source community. We utilize them strictly for non-commercial research purposes and ensure that our usage complies with the respective licenses and terms of use.

G.3 External Services

For automated evaluation, we utilize the Gemini 3 Pro API. We comply with the Google AI Studio Terms of Service regarding data handling and API usage limits.

Case Study: DeepSeek-R1-Distill-Qwen-1.5B on AIME

Question: Let ABC be a triangle inscribed in circle ω. Tangents at B and C intersect at D, and AD intersects ω at P. If $AB = 5$, $BC = 9$, $AC = 10$, find $m + n$ where $AP = m/n$.

Base Model (Failure): Calculation Anxiety & Spiral

Reasoning Trace:
[...Step 1: Coordinates setup $A(1/3, 4\sqrt{14}/3)$, $D(9/2, -27\sqrt{14}/11)$...]
[...Step 2: Derives quadratic for intersection...]
The equation is $218750(33 - 168t)^2/5544^2 + \cdots = 765/112$. ***The numbers are getting huge.*** 5544^2 ***is around 30 million. This doesn't seem right. Let me double check the slope of*** OB***. Did I miss a negative sign? Let me recalculate... Slope seems correct. Maybe I should use Stewart's Theorem instead? But I need length*** AD***... Let's go back to coordinates. Is it possible that*** y_A ***is negative? No... Let me try to factor 218750... No, this is taking too long.*** *[...Model abandons the correct coordinate path and loops between verifying arithmetic and trying new methods until cutoff...]*
Final Answer: Timeout / No Solution

GUARD (Success): Confidence Injection

Reasoning Trace:
[...Coordinates setup identical to Base Model...]
We arrive at the quadratic: $1521x^2 - 4314x + 1269 = 0$. The coordinates are large. I need to solve for x.
`\n\n` ***This...*** *[↑ Entropy Spike Detected after delimiter]*

⇒ GUARD Triggered: Branch 3 (Counterfactual) Selected
Let me reconsider: Instead of doubting the size, let's calculate the discriminant D carefully to see if it's a perfect square.
$D = b^2 - 4ac = 4314^2 - 4(1521)(1269)$. $4314^2 = 18,610,596$. $4(1521)(1269) = 6084 \times 1269 = 7,718,596$. $D = 18,610,596 - 7,718,596 = 10,892,000$.
Wait, $10,892,000 = 108920 \times 100$? No, 1089×10000. $\sqrt{D} = \sqrt{1089 \times 10000} = 33 \times 100 = 3300$. **It is a perfect square!**
Roots: $x = \frac{4314 \pm 3300}{3042}$. Solution P: $x = \frac{7614}{3042} = \frac{423}{169}$. Calculating distance AP: $AP = \sqrt{(423/169 - 1/3)^2 + \ldots} = \frac{100}{13}$.
Thus, $m = 100, n = 13$. $m + n = 113$.
Final Answer: 113

Analysis: This case illustrates an "Arithmetic Spiral." The Base model correctly derives the coordinate equations but panics due to the magnitude of the coefficients ($> 10^7$), interpreting the complexity as a sign of error (red text). It abandons the correct path to search for non-existent shortcuts. GUARD detects this hesitation and intervenes with a Counterfactual Branch that encourages verification rather than abandonment. By pushing through the discriminant calculation, the model discovers the numbers simplify beautifully (green text), resolving the problem.

Figure 7: **Qualitative Comparison on AIME.** The Base model (Top) retreats from a correct but computationally heavy solution at the structural boundary `\n\n`. GUARD (Bottom) detects the uncertainty spike immediately following the delimiter and steers the model to complete the calculation.

Case Study: DeepSeek-R1-Distill-Qwen-7B on Minerva

Question: Use the Bohr model of the atom to compute the wavelength of the transition from the $n = 100$ to $n = 99$ levels, in centimeters. [Useful relation: the wavelength of $L\alpha$ ($n = 2$ to $n = 1$ transition) is 1216Å].

Base Model (Failure): Intuition Trap & Unit Spiral

Reasoning Trace:
First, I need to find the Rydberg constant R. Using the $L\alpha$ transition ($n = 2 \to 1$): $1/\lambda = R(1/1^2 - 1/2^2) = 0.75R$. Given $\lambda = 1216\text{Å} = 1.216 \times 10^{-5}$ cm, so $R = 4/(3 \times 1.216 \times 10^{-5}) \approx 1.0965 \times 10^5$ cm^{-1}.
Now for the transition $n = 100 \to 99$: $1/\lambda' = R(1/99^2 - 1/100^2)$. $1/99^2 - 1/100^2 = \frac{10000-9801}{98010000} = \frac{199}{9.8 \times 10^7} \approx 2.03 \times 10^{-6}$. So $1/\lambda' = 1.0965 \times 10^5 \times 2.03 \times 10^{-6} \approx 0.222$ cm^{-1}. Therefore, $\lambda' = 1/0.222 \approx 4.49$ cm.
\n\n ***4.49 cm? That seems impossibly long. The L-alpha line is UV radiation (10^{-5} cm). How can an atomic transition be in the microwave range (centimeters)? Usually, atomic spectra are visible or UV.***
\n\n ***I must have messed up the units of the Rydberg constant. Let me check. Maybe 1216Å should be converted to meters first? 1.216×10^{-7} m. Re-calculating R in m^{-1}: $R \approx 1.097 \times 10^7 m^{-1}$. Plugging back: $1/\lambda' \approx 22.2m^{-1} \implies \lambda' \approx 0.045$ m. 0.045 m is still 4.5 cm. This doesn't make sense. Atoms are tiny, how can they emit centimeter-long waves?***
\n\n ***Let me try using the energy formula $E = -13.6\,eV/n^2$. $\Delta E = 13.6(1/99^2 - 1/100^2) \approx 2.7 \times 10^{-5}$ eV. Convert eV to wavelength... $12400/(2.7 \times 10^{-5}) \approx 4.5 \times 10^8$Å. That is HUGE. I must be wrong. Is it possible the question implies Z is not 1?*** *[...Model continues to loop through different constants (Planck, speed of light) trying to "fix" the large result, eventually hallucinating a scaling factor...]*
Final Answer: Invalid / Timeout

GUARD (Success): Scaling Law Verification

Reasoning Trace:
[...Initial calculation matches Base Model...] Using $R \approx 1.097 \times 10^5$ cm^{-1}, we derived $1/\lambda' \approx 0.222$ cm^{-1}. This yields $\lambda' \approx 4.49$ cm.
\n\n ***this result seems...*** *[↑ Entropy Spike Detected]*

⇒ GUARD Triggered: Branch 2 (Inhibitory) "Wait,"
Wait, let's not discard the result yet. Let's perform an order-of-magnitude estimation to see if this is physically plausible for high-n states (Rydberg states).
1. Energy Scaling: Energy levels scale as $1/n^2$. The transition $n = 2 \to 1$ involves a large gap: $(1-1/4) = 0.75$. The transition $n = 100 \to 99$ involves a tiny gap: $\approx 2/n^3 \approx 2/1,000,000 = 2 \times 10^{-6}$.
2. Ratio Comparison: Ratio of Energy gaps $\approx 0.75/(2 \times 10^{-6}) \approx 3.75 \times 10^5$. Since Wavelength $\lambda \propto 1/\Delta E$, the new wavelength should be roughly 3.75×10^5 times longer than $L\alpha$.
3. Verification: $\lambda_{L\alpha} = 1216\text{Å} \approx 1.2 \times 10^{-5}$ cm. Expected $\lambda' \approx (1.2 \times 10^{-5}\text{ cm}) \times (3.75 \times 10^5) \approx 4.5$ cm. This matches our calculated result of 4.49 cm. High-n transitions indeed produce low-energy photons (microwaves/radio waves). The calculation is correct.
Final Answer: **4.49 cm**

Analysis: The Base model enters an epistemic spiral triggered by the dissonance between its correct calculation (4.49 cm) and a faulty intuition. Driven by this misconception, the model misleads itself, engaging in a desperate search to prove its correct answer wrong. It repeatedly cycles through unit conversions and constant checks not to solve the problem, but to find a non-existent "bug" that would justify its intuition. GUARD interrupts this self-defeating loop by enforcing a physics-based scaling check, forcing the model to accept the counter-intuitive truth rather than spiraling into hallucination.

Figure 8: **Qualitative Comparison on Minerva.** The Base model (Top) correctly computes the value but actively rejects the result due to its counter-intuitive magnitude, entering a spiral to rationalize its error. GUARD (Bottom) breaks this self-defeating loop by injecting a scaling-law verification, confirming the physical validity of the result.

CASE STUDY: QWQ-32B ON OLYMPIADBENCH

Question: Define sequence $x_1 = a, x_{n+1} = 2x_n + 1$. Let $y_n = 2^{x_n} - 1$. Determine the largest k such that $y_1, \ldots, y_k$ are all prime for some positive integer a.

Base Model (Failure): The Empirical Spiral

Reasoning Trace:
Let's test small values of a. If $a = 2$: $x = \{2, 5, 11\}$, $y = \{3, 31, 2047\}$. $2047 = 23 \times 89$ (composite). So $k = 2$. If $a = 3$: $x = \{3, 7, 15\}$, $y = \{7, 127, 32767\}$. 32767 is composite. So $k = 2$.
\n\n ***Is it possible to find $k = 3$ with a larger a? We need x_1, x_2, x_3 to be Mersenne exponents. Let's check $a = 5$... $x_2 = 11$, $y_2 = 2047$ (composite). Fail. Let's check $a = 7$... $x_2 = 15$ (composite). Fail. Let's check $a = 13$... $x_2 = 27$ (composite). Fail.***
\n\n ***Maybe $a = 89$? y_1 is prime. $x_2 = 179$. Is $2^{179} - 1$ prime? That's too big to check mentally. Let me try finding patterns modulo 3. If $p_1 \equiv 1$, then $p_2 \equiv 0$ (bad). If $p_1 \equiv 2$, then $p_2 \equiv 2, p_3 \equiv 2$. This doesn't rule out $k = 3$. What about modulo 7? Modulo 5? I can't find a contradiction. Maybe $a = 127$? Or $a = 19937$? There must be some a that works...*** *[...Model spirals into an infinite search for a counter-example without a theoretical proof strategy...]*
Final Answer: 2 (Correct number, but low confidence / guessing)

GUARD (Success): Structural Proof Injection

Reasoning Trace:
[...Initial checks for $a = 2, 3$ match Base Model...]
For $a = 2$ and $a = 3$, we find $k = 2$. The chain breaks at the 3rd term (y_3) or 2nd term (y_2). I need to determine if $k = 3$ is possible for *any* a.
\n\n ***Let me assume $k = 3$...*** *[↑ Entropy Spike Detected]*

⇒ GUARD Triggered: Branch 3 (Counterfactual)) "Let me reconsider:"
Consider modulo 8. Let's pause the search and analyze the structure modulo 8. This might reveal a divisibility property.
Step 1: Parity Check. If $y_1 = 2^a - 1$ is prime, a must be prime. If $a = 2$ (even prime), we verified $k = 2$. Assume a is an *odd* prime.
Step 2: Modulo 8 Analysis. Sequence: $x_1 = a, x_2 = 2a + 1, x_3 = 4a + 3$. Since a is odd, $4a \equiv 4 \pmod 8$. Thus, $x_3 = 4a + 3 \equiv 4 + 3 \equiv 7 \pmod 8$.
Step 3: Quadratic Residue Argument. Since $x_3 \equiv 7 \pmod 8$, we know that 2 is a **quadratic residue** modulo x_3. $\Rightarrow \exists s$ such that $s^2 \equiv 2 \pmod{x_3}$.
Step 4: Fermat's Little Theorem. $2^{(x_3-1)/2} \equiv (s^2)^{(x_3-1)/2} \equiv s^{x_3-1} \equiv 1 \pmod{x_3}$. Note that the exponent $\frac{x_3-1}{2} = \frac{4a+2}{2} = 2a + 1 = x_2$. So, $2^{x_2} \equiv 1 \pmod{x_3}$.
Step 5: Conclusion. $2^{x_2} - 1 \equiv 0 \pmod{x_3}$. This means x_3 divides y_2. Since $y_2 > x_3$, y_2 must be composite. Therefore, for any odd prime a, the sequence breaks at y_2 (or earlier). The maximum length is $k = 2$ (achieved by $a = 2$).
Final Answer: 2

Analysis: The Base model falls into an "Empirical Spiral," trying to solve a structural number theory problem by brute-force enumeration. Lacking a proof strategy, it wanders through large candidates ($a = 89, 127 \ldots$) hoping to get lucky. GUARD detects the high entropy associated with this aimless search and intervenes with a Counterfactual Branch. By injecting the specific heuristic to "consider modulo 8," it guides the model to construct a rigorous proof using quadratic residues, transforming a guess into a mathematical certainty.

Figure 9: **Qualitative Comparison on OlympiadBench.** The Base model (Top) attempts to solve the problem by exhaustively checking prime candidates, eventually getting stuck in an infinite search space. GUARD (Bottom) detects the lack of convergence and steers the reasoning towards a structural proof using modular arithmetic and quadratic residues, proving that $k = 3$ is impossible.