DIAGNOSING HARMFUL CONTINUATION IN ANSWER-CORRECT MATHEMATICAL LONG-COT TRACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Long chain-of-thought (CoT) traces are widely used as supervision for reasoning-oriented LLM supervised finetuning (SFT), yet answer-correct traces can still lead to markedly different fine-tuning outcomes. We study post-conclusion continuation in answer-correct mathematical long-CoT data: a continuation where the answer appears sufficiently supported, but the trace continues with additional reasoning that remains in the supervised target. To test its training effect, we use a delete-only editor to construct answer-preserving suffix removal and compare CoT-based SFT on the original and processed traces. We observe improved SFT outcomes on mathematical problems after removing the editor-identified post-conclusion continuation, suggesting that this continuation is harmful to training in our setting. We therefore refer to this empirically supported phenomenon as *harmful continuation*. Beyond this intervention, we further characterize the removed post-conclusion continuation through uncertainty and hidden-state progress. We observe persistent local uncertainty together with weakened terminal-directional progress, forming an uncertainty–geometry mismatch. Finally, we introduce ***H**armful **C**ontinuation **C**ut (HCC)*, a lightweight proxy built on a frozen Qwen2.5-0.5B-Instruct backbone and trained to predict editor-identified boundaries with uncertainty–geometry auxiliary objectives. Our code is available at `anonymous.4open.science/r/HCC-76B3`.

1 INTRODUCTION

Long chain-of-thought traces have become important training targets for reasoning models (Wei et al., 2022; Luo et al., 2025; Ou & Yin, 2025). They are used not only in supervised fine-tuning Ou & Yin (2025), but also in reasoning-oriented continued training and as cold-start data before reinforcement learning Wang et al. (2026). Unlike final-answer annotations, long-CoT traces expose full reasoning trajectories that models are encouraged to imitate during training. This makes CoT trace quality a central issue for reasoning training where the training target specifies not only what answer to produce, but also what trajectory should be treated as learnable reasoning behavior.

Prior work has already shown that answer-correct CoT traces can differ substantially in training utility. The style and structure of source trajectories can affect the generalization of SFT models (Tian et al., 2025; Zhang et al., 2026; Li et al., 2025), and recent studies further connect trace compatibility, reasoning patterns, and learnability to downstream outcomes (Liu et al., 2026; Li et al., 2026). However, most existing methods remain confined to trace selection, prefix selection, or externally guided rewriting, which leaves the internal failure mode of answer-correct traces under-explained. As a result, they do not characterize where useful reasoning may end and begin, or why such a phase can be associated with weaker SFT despite preserving answer correctness.

To address this gap, we take a diagnostic view of answer-correct mathematical long-CoT training traces. We seek a trace-internal explanation for why answer-correct mathematical reasoning traces may differ in training utility, rather than assuming a long reasoning trace is uniformly useful once its final answer is correct. We do not claim that every long tail is harmful or treat length as the central issue. Instead, we ask whether some traces enter a low-value post-conclusion continuation: the answer is already sufficiently supported, but subsequent reasoning remains locally costly while showing weak hidden-state progress. From the uncertainty perspective, we observe that some post-conclusion continuation remains locally costly or unstable, suggesting that the trace continues to

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

explore after evaluator-based answer support has largely saturated. From the geometric perspective, this continued exploration shows weakened terminal-directional hidden-state progress. We refer to this hypothesized low-value phase as post-conclusion continuation before evaluating its downstream training effect. When answer-preserving removal of this continuation improves SFT outcomes, we call the empirically supported training-unfavorable case harmful continuation.

To examine the training relevance of this post-conclusion continuation, we use a delete-only editor as an operational intervention tool. The editor does not rewrite the trace; it only removes post-conclusion suffixes while preserving the original prefix and final answer. This allows us to test whether answer-preserving removal of the post-conclusion continuation improves SFT outcomes. Motivated by this diagnosis, we instantiate ***H**armful **C**ontinuation **C**ut (HCC)*, a lightweight boundary proxy. HCC uses a frozen Qwen2.5-0.5B-Instruct backbone with a cut head to extract sentence-level reasoning states and approximate the editor-identified post-conclusion continuation.

Our contributions are summarized as follows:

- We define harmful continuation in mathematical long-CoT traces and show that its removal improves SFT in our settings.
- We reveal an uncertainty–geometry mismatch in the removed continuation.
- We propose HCC, a lightweight proxy that approximates editor-identified boundaries and their downstream benefits.

2 Related Work

Long-CoT reasoning training. Long-CoT traces have become important for post-training pipelines. While early pipelines often treat answer-correct traces as directly usable supervision, recent studies show that correctness alone does not determine their training value. Work on data selection, trace compatibility, and informative alignment Yang et al. (2026); Zhang et al. (2026); Chandra et al. (2025) suggests that only a subset of answer-correct traces provides beneficial supervision. Other approaches modify reasoning traces through sequence truncation, prefix optimization, adaptive prefix alignment, robustness to partial reasoning, or length-aware training Chen et al. (2025a); Sun et al. (2026); Liu et al. (2026); Silvestri & Cetin (2026); Xu et al. (2025); Luo et al. (2026); Ma et al. (2025). However, these methods mainly operate in a heuristic manner. They do not directly characterize where useful reasoning may end, where useless reasoning may begin, or why this reasoning can be associated with weaker SFT supervision despite answer correctness.

Properties of long-CoT trajectories. Recent work increasingly treats long-CoT traces as structured reasoning trajectories rather than flat text sequences. Studies on overthinking show that long-reasoning models may repeatedly verify intermediate conclusions or continue reasoning without meaningful gain Chen et al. (2025b). Complementary analyses examine global reasoning patterns, trajectory geometry, and step-level anchors that contribute to actual progress Jiang et al. (2025); Ballon et al. (2026); Bogdan et al.; Yang et al. (2025). Most closely related to our motivation, Li et al. (2026) connects trajectory properties to downstream SFT outcomes and shows that different reasoning patterns can lead to different generalization behavior. Our work builds on this trajectory-level view, but focuses on a post-conclusion continuation inside answer-correct traces: continuation that remains uncertain or costly while showing weakened terminal-directional progress.

3 Operational Partition and Diagnostics of Post-Conclusion Continuation

3.1 Data Construction

In this section, we do not assume that editor-removed sentences are ground-truth harmful continuation. Instead, we use the delete-only editor to construct an operational partition for diagnosing post-conclusion suffixes. The resulting groups are used to reveal statistical signatures of a possible low-value phase, rather than to define harmfulness by editor decisions alone. We use Qwen3-235B-A22B-Instruct-2507 Team (2025) and DeepSeek-R1-V3.2 Guo et al. (2025) to generate trajectories, and sample 4,780 answer-correct long-CoT solution trajectories from the OpenR1-Math-220k

dataset. These trajectories serve as the original CoT training traces. For simplicity, we use $\{T\}_Q$ and $\{T\}_R$ to refer to the two sets of trajectories from the two models, respectively. We then employ Qwen3.5-27B Team (2025) as a delete-only offline editor to expose post-conclusion continuation for empirical analysis. Given a trajectory from $\{T\}_Q$ or $\{T\}_R$, the editor marks the post-conclusion sentences that can be removed while preserving the reasoning necessary to recover the final answer.

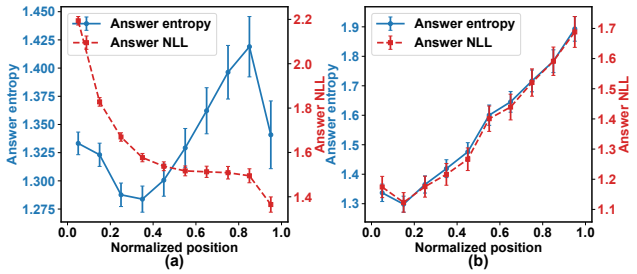


Figure 1: Evaluator-based uncertainty diagnostics as reasoning segments are progressively added for (a) retained reasoning and (b) editor-removed continuation.

Operational groups. We divide each edited trajectory into operational groups: (1) *retained reasoning*, is the editor-preserved portion supporting the final answer. (2) *editor-removed continuation*, is the post-conclusion continuation marked as removable by the offline editor. These terms here denote operational groups rather than predefined theoretical labels.

3.2 Uncertainty View

In this section, we aim to answer the following question: *Does the post-conclusion continuation continue to improve evaluator-based final-answer recoverability, or does answer support appear to saturate while local uncertainty remains high?*

Comparison protocols. We analyze uncertainty at both answer and sentence levels. At the answer level, we progressively append reasoning sentences along the same complete response trajectory and compute prefix-conditioned final-answer entropy and negative log-likelihood (NLL) Malinin & Gales (2020). These quantities are interpreted as evaluator-based diagnostics of answer recoverability, rather than direct measurements of causal reasoning contribution. For segment-wise visualization, positions are normalized separately within retained reasoning and the subsequent editor-removed continuation. At the sentence level, we use sentence entropy and sentence NLL to measure local predictive difficulty. For boundary-level analysis, we track K_1, K_T, C_1, and C_T, denoting the first and last sentences of retained reasoning and editor-removed continuation, and compare both local uncertainty changes and answer-NLL reduction. The protocols are described in the Appendix C.

Answer uncertainty dynamics. Figure 1 shows how answer-level uncertainty changes as reasoning sentences are progressively appended to the same complete response. For visualization, the x-axis is normalized within its corresponding segment, with Figure 1 (a) showing retained reasoning and Figure 1 (b) showing the subsequent editor-removed continuation. In retained reasoning, answer entropy changes non-monotonically but does not exhibit a persistent increase, while answer NLL steadily decreases as more useful reasoning is added. This suggests that the retained segment improves evaluator-based final-answer recoverability even when intermediate reasoning involves local exploration or verification. In contrast, once the trace enters editor-removed continuation, both answer entropy and answer NLL increase as more post-conclusion content is appended. This suggests that the continuation does not consistently improve evaluator-based answer recoverability, but instead introduces a higher-uncertainty state after the answer has been sufficiently supported.

Boundary-level mismatch. Figure 2 examines local uncertainty and evaluator-based answer-support changes on the boundary between retained reasoning and editor-removed continuation. From K_1 to K_T, sentence entropy and sentence NLL increase, but the answer-NLL reduction also becomes stronger, suggesting that local uncertainty within retained reasoning can still accompany improved answer recoverability under the evaluator. The transition from K_T to C_1 shows a different pattern. Local uncertainty rises at the beginning of editor-removed continuation, while the gain in answer support no longer increases correspondingly. From C_1 to C_T, this high-uncertainty regime is

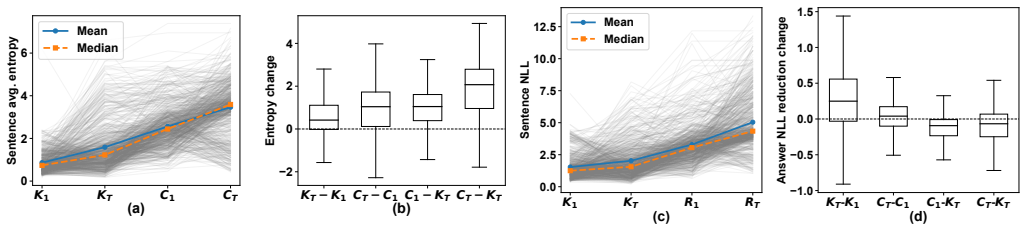


Figure 2: Boundary-level diagnostic changes around the editor-identified post-conclusion continuation: (a) sentence entropy, (b) entropy change, (c) sentence NLL, and (d) answer-NLL reduction. maintained or amplified, but the continuation does not provide stable additional answer-NLL reduction. The candidate low-value pattern therefore emerges when increased local prediction difficulty is no longer matched by consistent improvements in evaluator-based final-answer recoverability.

3.3 Geometric View

Following the uncertainty analysis, another question arises: *Does the increased predictive uncertainty of post-conclusion continuation translate into effective hidden-state progress?*

Comparison protocols. Following prior work on the geometry of Transformer hidden representations Valeriani et al. (2023); Gurnee & Tegmark (2024) and trajectory-level analyses of long-CoT reasoning Jiang et al. (2025), we use sentence-boundary hidden states as an operational proxy for reasoning-state evolution. Specifically, hidden displacement measures the magnitude of representation change between consecutive reasoning steps, while forward progress measures the component of this change aligned with the terminal direction of the analyzed trace. These metrics characterize representation-level state movement and terminal-directional progress under an operational proxy. Because the terminal direction is derived from the observed trace representation, forward progress should be interpreted as an operational terminal-directional proxy rather than a ground-truth answer direction. We further compute progress efficiency, defined as the ratio between forward progress and hidden displacement, to measure how effectively local state movement is converted into directional progress. To control for sentence length, we also report token-normalized variants of hidden displacement and forward progress. Curvature is included as an auxiliary diagnostic of directional change, while displacement, forward progress, and efficiency serve as the primary geometric indicators. The detailed protocols are described in Appendix.

Distributional geometric tendency. Figure 3(a) shows that many sentences in both groups have near-zero token-normalized hidden displacement, suggesting that fine-grained geometric scores should not be used as hard sentence-level deletion criteria. Nevertheless, retained reasoning is shifted toward larger displacement, indicating stronger state movement per token than editor-removed continuation. Figure 3(b) shows a similar pattern. Although the two groups overlap substantially in the scatter space, retained reasoning more often exhibits larger hidden displacement and stronger forward progress. Thus, geometry provides a distributional signal of useful reasoning progress rather than a pointwise separation rule.

Paired evidence. Table 1 gives a paired per-sample comparison and shows that editor-removed continuation has weaker operational hidden-state progress than retained reasoning. For hidden displacement, the removed mean is much lower than the retained mean (21.91 vs. 44.92), and the paired difference is consistently negative, with 79% of samples showing lower values for the removed segment. Forward progress shows the same trend, suggesting that the removed continuation advances less under the terminal-directional proxy. These gaps remain after token normalization, where both hidden displacement per token and forward progress per token are lower for editor-removed continuation. By contrast, curvature has only a small absolute gap, suggesting that the post-conclusion continuation is associated mainly with weaker displacement and weaker terminal-directional progress, rather than with a curvature pattern.

3.4 Uncertainty–Geometry Mismatch

The above analyses reveal a consistent diagnostic pattern in the editor-identified post-conclusion continuation. From the uncertainty view, this suffix often remains locally costly or unstable, while evaluator-based answer recoverability no longer improves consistently. Geometrically, the same suf-

fix shows weaker hidden displacement and weaker terminal-directional progress than retained reasoning. We refer to this pattern as an uncertainty–geometry mismatch. Importantly, this mismatch does not define harmfulness and does not itself prove causal training harm. Instead, it characterizes the editor-removable post-conclusion continuation later tested through downstream SFT intervention. Additional cross-editor and positional controls in Appendix D.2 show substantial boundary agreement and lower per-token progress for removed continuations than for the later answer segment, suggesting that neither editor choice nor late position alone explains the diagnostic pattern.

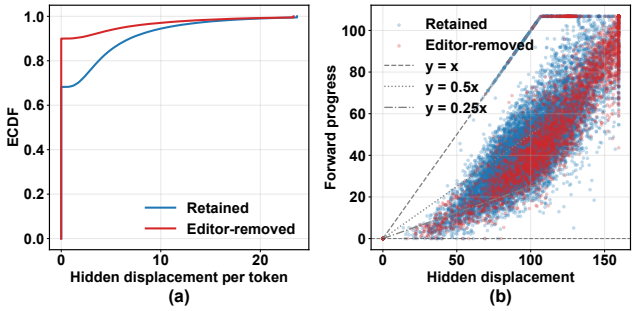


Figure 3: Hidden-state progress of retained reasoning and editor-removed continuation: (a) ECDF of token-normalized hidden displacement and (b) hidden displacement versus forward progress.

Table 1: Paired per-sample comparison of operational hidden-state progress between editor-removed continuation and retained reasoning. The paired difference is defined as Δ = Removed mean − Retained mean. Removed lower denotes the fraction of samples with $\Delta < 0$.

Metric	Removed mean	Retained mean	Removed lower	95% CI of Δ
Hidden displacement	21.91	44.92	0.79	$[-23.85, -22.19]$
Forward progress	10.79	20.50	0.79	$[-10.15, -9.31]$
Progress efficiency	0.09	0.19	0.79	$[-0.11, -0.10]$
Hidden displacement / token	1.65	2.83	0.79	$[-1.24, -1.11]$
Forward progress / token	0.84	1.28	0.79	$[-0.47, -0.40]$
Curvature	1.09	1.17	0.73	$[-0.09, -0.08]$

4 METHOD

Our diagnostic analysis suggests that editor-removed post-conclusion continuation is associated with persistent local uncertainty and weak terminal-directional hidden-state progress. Based on this operational pattern, we instantiate a lightweight boundary proxy termed Harmful Continuation Cut.

4.1 BOUNDARY PROXY

Given a question q and a verified source trajectory, we separate the reasoning trace from the final answer. Let $r = (r_1, r_2, \ldots, r_T)$ denote the reasoning trace and a^* the final answer. Let $c^* \in \{0, \ldots, T\}$ denote the editor-identified post-conclusion continuation boundary used for supervision. Our goal is to learn a lightweight proxy that predicts this boundary, so that the retained prefix $r_{\le c^*}$ is kept while the post-conclusion continuation $r_{>c^*}$ is removed. We encode the question and reasoning trace with a frozen causal language model and extract a hidden representation at each sentence boundary, obtaining $h_t \in \mathbb{R}^D$. This representation denotes the model state after consuming the prefix up to sentence r_t. We pass the sentence-level states through a shared sequence encoder:

$$\tilde{h}_t = \mathrm{SeqEnc}(h_{1:T})_t. \tag{1}$$

The contextualized representation $\tilde{h}_t$ is used as the common input for latent regularization and uncertainty–geometry diagnostic estimation.

4.2 SEQUENTIAL LATENT REGULARIZATION

HCC uses a sequential variational latent representation to regularize sentence-level boundary states. For each contextualized sentence state $\tilde{h}_t$, we define a posterior latent distribution:

$$q_\phi(z_t \mid \tilde{h}_t) = \mathcal{N}(\mu_t, \Sigma_t), \tag{2}$$

where the mean and variance are predicted from $\tilde{h}_t$. We also define a sequential prior from the previous contextual state:

$$p_\eta(z_t \mid \tilde{h_{t-1}}) = \mathcal{N}(\mu_t^p, \Sigma_t^p). \tag{3}$$

The sampled latent variable is projected back to the boundary-prediction space:

$$b_t = f_{\text{lat}}(z_t). \tag{4}$$

The latent representation is regularized by:

$$\mathcal{L}_{\text{KL}} = \sum_{t=1}^{T} D_{\text{KL}} \left(q_\phi(z_t \mid \tilde{h}_t) \,\|\, p_\eta(z_t \mid \tilde{h_{t-1}}) \right). \tag{5}$$

This term provides latent regularization for boundary prediction, rather than a hard information bottleneck or an explicit answer-prediction objective.

4.3 Uncertainty–Geometry Diagnostic Estimation

Post-conclusion continuation may remain locally costly or unstable while contributing limited hidden-state progress. To capture this uncertainty–geometry mismatch, HCC jointly estimates a local uncertainty and an operational progress signal from $\tilde{h}_t$. For uncertainty, we define a scalar target T_t from source-trace sentence-level statistics such as entropy, NLL, or log-perplexity. HCC estimates this signal as:

$$s_t^{\text{ent}}, \hat{T}_t = f_{\text{ent}}(\tilde{h}_t), \tag{6}$$

where s_t^{ent} is the uncertainty-aware context vector used for boundary prediction, and $\hat{T}_t$ is the scalar uncertainty estimate. The regression loss is:

$$\mathcal{L}_{\text{ent}} = \sum_{t=1}^{T} \text{Huber}(\hat{T}_t, T_t). \tag{7}$$

For geometry, we define a scalar progress target G_t from hidden-state movement statistics. HCC estimates this signal as:

$$s_t^{\text{geo}}, \hat{G}_t = f_{\text{geo}}(\tilde{h}_t), \tag{8}$$

where s_t^{geo} is the progress-aware context vector used for boundary prediction, and $\hat{G}_t$ is the scalar progress estimate. The regression loss is:

$$\mathcal{L}_{\text{geo}} = \sum_{t=1}^{T} \text{Huber}(\hat{G}_t, G_t). \tag{9}$$

Together, these estimates provide a unified diagnostic representation of whether a post-conclusion sentence remains locally uncertain while showing weak operational progress.

4.4 Mismatch-Aware Boundary Prediction

HCC fuses the latent regularization signal with the uncertainty–geometry diagnostic representation in a shared boundary-prediction space:

$$m_t = \text{LN}\left(b_t + \alpha_{\text{geo}} s_t^{\text{geo}} + \alpha_{\text{ent}} s_t^{\text{ent}} \right), \tag{10}$$

where α_{geo} and α_{ent} are learnable scalar gates. Thus, HCC does not concatenate the raw contextual state directly into the final cut representation. Instead, the scalar estimates $\hat{T}_t$ and $\hat{G}_t$ are trained with auxiliary losses, while their context vectors contribute to boundary prediction.

For cut prediction, we prepend a learned beginning-of-sequence state m_0 and compute:

$$\pi_t = \text{CutHead}(m_t), \quad t = 0, \ldots, T. \tag{11}$$

Here, π_t denotes the logit of choosing sentence t as the last retained sentence. Given the editor-identified boundary c^*, the cut head is trained with:

$$\mathcal{L}_{\text{cut}} = -\log \frac{\exp(\pi_{c^*})}{\sum_{j=0}^{T} \exp(\pi_j)}. \tag{12}$$

We also train a sentence-level deletion head to match the deletion labels produced by the offline editor. Let $y_t \in \{0, 1\}$ denote whether sentence r_t should be deleted. The deletion probability is:

$$\hat{y}_t = \mathrm{DelHead}(m_t), \tag{13}$$

with the binary deletion loss:

$$\mathcal{L}_{\mathrm{del}} = -\sum_{t=1}^{T} \left[y_t \log \hat{y}_t + (1 - y_t) \log(1 - \hat{y}_t) \right]. \tag{14}$$

The overall training objective is:

$$\mathcal{L} = \mathcal{L}_{\mathrm{cut}} + \lambda_{\mathrm{del}} \mathcal{L}_{\mathrm{del}} + \lambda_{\mathrm{KL}} \mathcal{L}_{\mathrm{KL}} + \lambda_{\mathrm{ent}} \mathcal{L}_{\mathrm{ent}} + \lambda_{\mathrm{geo}} \mathcal{L}_{\mathrm{geo}}. \tag{15}$$

4.5 Architecture and rationale.

HCC uses a frozen Qwen2.5-0.5B-Instruct backbone and a sequence encoder to obtain contextualized sentence-boundary states. A sequential latent branch, uncertainty–geometry estimators, and cut/deletion heads are supervised by $\mathcal{L}_{\mathrm{KL}}$, $\{\mathcal{L}_{\mathrm{ent}}, \mathcal{L}_{\mathrm{geo}}\}$, and $\{\mathcal{L}_{\mathrm{cut}}, \mathcal{L}_{\mathrm{del}}\}$, respectively. Together, these objectives are designed to combine state regularization, diagnostic supervision, and direct boundary supervision, reducing reliance on source-specific boundary cues. When processing new traces, HCC predicts the cut without the editor, and neither HCC nor the editor is required by the downstream student at inference time.

5 Experiments

5.1 Experimental Setup

Datasets. We use the same trajectories as in Section 3, also denoted as $\{T\}_Q$ and $\{T\}_R$. For HCC training, we use the `Qwen2.5-0.5B-Instruct` as a frozen backbone and train lightweight prediction heads to learn the editor-identified removable post-conclusion continuation boundary. To test whether the learned deletion rule transfers across nearby source-model families rather than only memorizing a single source style, we construct the train–validation split across source models. Specifically, we use $\{T\}_Q$ to train HCC, then process $\{T\}_R$ with the trained HCC and SFT the baseline model on the processed $\{T\}_R$, and vice versa.

Evaluation metrics. We include 4 different methods to process the original trajectories, including (1) Vanilla, indicating that the original trajectories are directly used for SFT without any processing; (2) Editor, indicating that the trajectory is processed by Qwen3.5-27B to remove the post-conclusion continuation; (3) Heuristic, indicating the heuristic method proposed by Li et al. (2026) and (4) HCC, indicating the proposed method in this paper. The detailed process of the 4 methods is described in Appendix A.2. We evaluate the resulting SFT models on three reasoning benchmarks: MATH500, AMC23, and GSM8K. The primary evaluation metric is pass@1.

5.2 Main Results

Overall comparisons. Table 2 reports the main SFT results using different processed versions of the same answer-correct long-CoT traces. From the results, we can draw several conclusions: (1) Across different training traces and models, answer-preserving removal of the editor-identified post-conclusion continuation improves downstream SFT outcomes over training on the original traces. This provides interventional evidence that the post-conclusion continuation is training-unfavorable in these settings. (2) HCC achieves performance close to the 27B editor-processed reference, and in several cases even surpasses it. For example, in the LLaMA3.2-3B setting, HCC obtains an average score of 45.2 on $\{T\}_Q$ and 49.3 on $\{T\}_R$, closely matching the editor-processed results of 45.7 and 49.8. HCC also outperforms the editor reference on MATH500 under $\{T\}_Q$ and on GSM8K under $\{T\}_R$. These results indicate that a lightweight boundary proxy can recover much of the benefit of large-model delete-only trace editing. (3) Compared with heuristic method, HCC-processed traces are both more accurate and substantially shorter in response length. The comparison with heuristic method suggests that length reduction alone may not fully explain the gains, although length-controlled interventions would be needed for a complete separation. (4) HCC yields weaker

and less consistent gains on Qwen2.5-Math-7B. This may be because the model has already undergone SFT on high-quality mathematical data; Appendix D.5 reports corresponding results for the Qwen2.5-7B without prior math-specific SFT.

Table 2: Comparison of different training trajectories. Len. denotes the average response length.

Method	SFT on $\{T\}_Q$					SFT on $\{T\}_R$				
	MATH500	**AMC23**	**GSM8K**	**Avg.**	**Len.**	**MATH500**	**AMC23**	**GSM8K**	**Avg.**	**Len.**
Backbone: LLaMA3.2-3B-Instruct										
Vanilla	29.8	10.0	69.0	36.3	3478.8	38.4	15.0	69.9	41.1	4967.1
Heuristic	34.0	15.0	71.5	40.2	3430.8	40.0	15.0	70.1	41.7	5172.1
Editor	41.8	**20.0**	**75.4**	**45.7**	1906.1	**51.2**	**22.5**	75.9	**49.8**	1942.1
HCC	**43.2**	17.5	75.1	45.2	2010.1	47.6	**22.5**	**77.8**	49.3	1934.4
Backbone: Qwen2.5-Math-7B-Instruct										
Vanilla	**85.8**	62.5	95.8	81.4	519.6	85.6	62.5	95.8	81.3	510.6
Heuristic	82.8	57.5	95.8	78.7	526.1	84.4	62.5	95.7	80.9	498.5
Editor	84.4	**70.0**	**96.2**	**83.5**	499.0	86.4	**67.5**	96.0	**83.3**	443.2
HCC	82.6	62.5	96.1	80.4	497.9	**86.6**	65.0	**96.2**	82.6	455.9

Table 3: Ablation study of loss components on LLaMA3.2-3B-Instruct trained with processed $\{T\}_R$.

No.	λ_{del}	λ_{KL}	λ_{ent}	λ_{geo}	**ACC@8** $\uparrow$
0	0.0	0.0	0.0	0.0	21.5
1	0.0	0.0	0.4	0.4	46.9
2	0.4	0.4	0.0	0.0	38.5
3	0.8	0.8	0.4	0.4	**72.3**

Table 4: Comparison for LLaMA3.2-3B-Instruct SFT on $\{T\}_R$. Random Span preserves the final answer and retains a randomly located contiguous reasoning span matching HCC's per-trace retained length.

Method	**MATH500**	**AMC23**	**GSM8K**	**Avg.**
Random Span	31.4	7.5	48.1	29.0
HCC	47.6	22.5	77.8	49.3

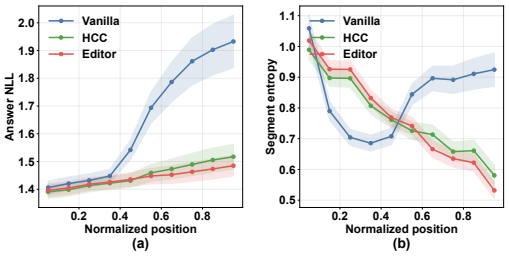


Figure 4: Post-training uncertainty diagnostics for generated reasoning traces.

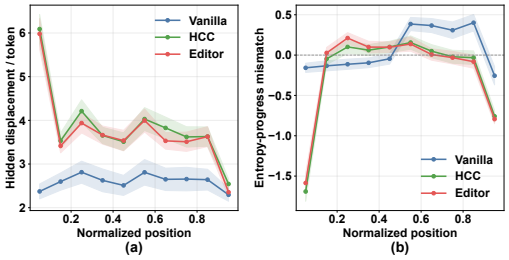


Figure 5: Operational uncertainty-progress diagnostics in generated reasoning traces.

5.3 More Experiments

Ablation Study We examine the contribution of HCC's auxiliary objectives under cross-source boundary prediction. We train HCC on $\{T\}_R$ traces and evaluate it on $\{T\}_Q$ traces using ACC@8, defined as the percentage of predicted cuts within eight sentences of the editor-identified boundary. As shown in Table 3, we can see that: (1) The cut loss provides only one boundary label per trace and may therefore rely on source-specific positional or stylistic cues. (2) The deletion loss supplies dense sentence-level supervision, while the sequential KL loss regularizes the evolution of neighboring latent states. (3) The uncertainty and geometry objectives further anchor these representations to the diagnostic mismatch associated with post-conclusion continuation. Together, these objectives couple boundary localization with its operational diagnostic structure, providing a more transferable inductive bias across trajectory sources.

Analysis of uncertainty dynamics. Figure 4 compares the reasoning dynamics of models trained on original, HCC-processed, and editor-processed traces. In Figure 4 (a), Vanilla shows a sharp late-stage increase in answer NLL, suggesting that generated reasoning becomes less favorable under

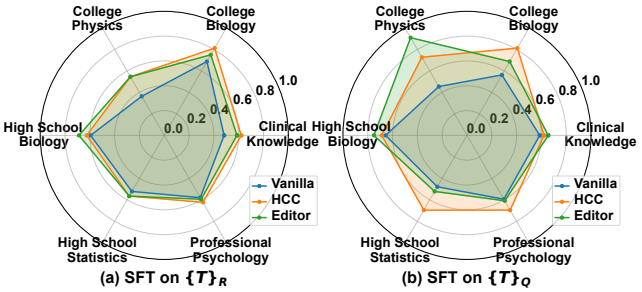


Figure 6: Visualization of the performance of LLaMA3.2-3B-Instruct on selected MMLU subjects.

evaluator-based answer recoverability. In contrast, both HCC and Editor keep answer NLL much more stable across the reasoning process, suggesting that post-conclusion continuation removal improves answer-support consistency under this diagnostic. Figure 4 (b) further shows the segment-level uncertainty pattern. Vanilla first reduces entropy but then exhibits a clear entropy rebound in the later reasoning bins, which is consistent with re-entering an unstable post-conclusion continuation. By comparison, HCC and Editor continue to reduce segment entropy in the later stage and produce highly similar curves. These curves provide post-training diagnostic evidence consistent with Section 3 pattern. The similarity between HCC and Editor curves suggests that the lightweight proxy approximates the diagnostic behavior of delete-only edited traces.

Analysis of geometric dynamics. Figure 5 examines whether processed traces lead to stronger hidden-state progress under the chosen proxy. In Figure 5 (a), HCC and Editor produce larger token-normalized hidden displacement than Vanilla, suggesting stronger state movement per generated token. Figure 5 (b) shows that Vanilla exhibits a clearer positive entropy-progress mismatch in the middle and late stages, where uncertainty is not matched by sufficient geometric progress. By contrast, HCC and Editor keep this mismatch closer to zero and reduce it near the end. These curves provide post-training diagnostic evidence consistent with Section 3 pattern, and again suggest that HCC approximates the delete-only editor behavior.

Analysis of Random Span. We use Random Span to test whether HCC's gains arise from retaining the same trajectory length. For trace i, x_i denotes HCC's retained reasoning length. Random Span preserves the final answer and retains a randomly located contiguous reasoning span of length x_i from the trace. As shown in Table 4, Random Span obtains an average of 29.0, compared with 49.3 for HCC. Because this control matches HCC's retained length for every trace while breaking the correspondence between retained content and the predicted boundary, the gap indicates that per-trace length matching alone is insufficient. HCC instead uses a trace-specific boundary to preserve the answer-supporting prefix and remove the identified post-conclusion continuation.

Capability preservation on MMLU. Since HCC modifies supervision from mathematical traces, we use six MMLU subjects Hendrycks et al. (2021) to check for degradation of previously acquired capabilities. As shown in Figure 6, HCC-based SFT consistently outperforms Vanilla-based SFT and remains comparable to the editor-based reference across the selected subjects. This comparison indicates that the improvements obtained from HCC-processed mathematical traces are not accompanied by a trade-off in performance on these knowledge-intensive evaluations.

6 Conclusion

We study post-conclusion continuation in answer-correct mathematical long-CoT traces without presuming it harmful. Answer-preserving removal by a delete-only editor improves SFT in our settings, supporting the identified continuation as training-unfavorable. We further associated the removed continuation with an uncertainty–geometry mismatch and developed HCC as a lightweight proxy for editor-identified removal boundaries. Results across different data and models support HCC as a practical approximation to editor-based processing within mathematical long-CoT SFT.

AI USE STATEMENT

In this work, we used generative AI tools in several capacities. First, Qwen3-235B-A22B-Instruct-2507 and DeepSeek-R1-V3.2 were used to generate mathematical long-CoT trajectories, and Qwen3.5-27B was used as a delete-only editor to construct the operational continuation boundaries and processed training traces described in Section 3.1. Second, ChatGPT-5.5 was used to draft portions of the manuscript based on technical content provided by the authors. Third, it was used to improve the readability, concision, and grammatical quality of author-written text. Finally, ChatGPT-5.5 was used for code editing. All AI-assisted manuscript content was reviewed and revised by the authors. The generated trajectories were filtered for answer correctness, and the editor outputs were treated as operational labels rather than ground-truth boundaries and evaluated through the controls reported in the paper. The authors verified the technical claims, numerical results, citations, and conclusions and take full responsibility for the final content of this work.

ETHICS STATEMENT

This work analyzes model-generated mathematical reasoning traces and does not involve human subjects or personally identifiable information. The datasets and models are used subject to their applicable licenses. The authors have read and adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We support reproducibility by documenting the trajectory construction and operational partition in Section 3.1, the HCC architecture and training objectives in Section 4, and the datasets, comparison methods, and evaluation protocol in Section 5. The Appendix provides the SFT and evaluation settings, detailed definitions of the comparison methods in Appendix A.2, and the uncertainty and geometric diagnostic protocols in Appendix C. The anonymized implementation is available at `https://anonymous.4open.science/r/HCC-76B3`.

REFERENCES

Marthe Ballon, Brecht Verbeken, Vincent Ginis, and Andres Algaba. Probing the trajectories of reasoning traces in large language models. *arXiv preprint arXiv:2601.23163*, 2026.

Paul C Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. Thought anchors: Which llm reasoning steps matter? In *Mechanistic Interpretability Workshop at NeurIPS 2025*.

Abhranil Chandra, Ayush Agrawal, Arian Hosseini, Sebastian Fischmeister, Rishabh Agarwal, Navin Goyal, and Aaron Courville. Shape of thought: When distribution matters more than correctness in reasoning tasks. *arXiv preprint arXiv:2512.22255*, 2025.

Wei-Rui Chen, Vignesh Kothapalli, Ata Fatahibaarzi, Hejian Sang, Shao Tang, Qingquan Song, Zhipeng Wang, and Muhammad Abdul-Mageed. Distilling the essence: Efficient reasoning distillation via sequence truncation. *arXiv preprint arXiv:2512.21002*, 2025a.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of long reasoning models. In *Forty-second International Conference on Machine Learning*, 2025b.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *International Conference on Learning Representations*, volume 2024, pp. 2483–2503, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Gangwei Jiang, Yahui Liu, Zhaoyi Li, Wei Bi, Fuzheng Zhang, Linqi Song, Ying Wei, and Defu Lian. What makes a good reasoning chain? uncovering structural patterns in long chain-of-thought reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6501–6525, 2025.

I Lawrence and Kuei Lin. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, pp. 255–268, 1989.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25366–25394, 2025.

Zhaoyi Li, Xiangyu Xi, Zhengyu Chen, Wei Wang, Gangwei Jiang, Ranran Shen, Linqi Song, Ying Wei, and Defu Lian. On the role of reasoning patterns in the generalization discrepancy of long chain-of-thought supervised fine-tuning. *arXiv preprint arXiv:2604.01702*, 2026.

Zhenghao Liu, Zhuoyang Wu, Xinze Li, Yukun Yan, Shuo Wang, Zulong Chen, Yu Gu, Ge Yu, and Maosong Sun. Long-chain reasoning distillation via adaptive prefix alignment. *arXiv preprint arXiv:2601.10064*, 2026.

Haotian Luo, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Xiaochun Cao, Dacheng Tao, Naiqiang Tan, and Li Shen. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 14242–14257, 2026.

Yijia Luo, Yulin Song, Xingyao Zhang, Jiaheng Liu, Weixun Wang, GengRu Chen, Wenbo Su, and Bo Zheng. Deconstructing long chain-of-thought: A structured reasoning optimization framework for long cot distillation. *arXiv preprint arXiv:2503.16385*, 2025.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. Cot-valve: Length-compressible chain-of-thought tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6025–6035, 2025.

Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*, 2020.

Linyu Ou and YuYang Yin. Empowering lightweight mllms with reasoning via long cot sft. *arXiv preprint arXiv:2509.03321*, 2025.

Gianluigi Silvestri and Edoardo Cetin. Learning from partial chain-of-thought via truncated-reasoning self-distillation. *arXiv preprint arXiv:2603.13274*, 2026.

Yiliu Sun, Zicheng Zhao, Yang Wei, Yanfang Zhang, and Chen Gong. Well begun, half done: Reinforcement learning with prefix optimization for llm reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33144–33152, 2026.

Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

Xiaoyu Tian, Yunjie Ji, Haotian Wang, Shuaiting Chen, Sitong Zhao, Yiping Peng, Han Zhao, and Xiangang Li. Not all correct answers are equal: Why your distillation source matters. *arXiv preprint arXiv:2505.14464*, 2025.

Lucrezia Valeriani, Diego Doimo, Francesca Cuturello, Alessandro Laio, Alessio Ansuini, and Alberto Cazzaniga. The geometry of hidden representations of large transformer models. *Advances in Neural Information Processing Systems*, 36:51234–51252, 2023.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Advances in Neural Information Processing Systems*, 38:115452–115486, 2026.

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

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

Shiming Yang, Yuxuan Tong, Xinyao Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning. In *Forty-second International Conference on Machine Learning*, 2025.

Yuming Yang, Mingyoung Lai, Wanxu Zhao, Xiaoran Fan, Zhiheng Xi, Mingqi Wu, Chiyue Huang, Jun Zhao, Haijun Lv, Jian Tong, et al. Which reasoning trajectories teach students to reason better? a simple metric of informative alignment. *arXiv preprint arXiv:2601.14249*, 2026.

Dylan Zhang, Qirun Dai, and Hao Peng. The best instruction-tuning data are those that fit. *Advances in Neural Information Processing Systems*, 38:141172–141208, 2026.

A Implementation Details

A.1 Training Settings

We implement SFT with LLaMAFactory. All models are fine-tuned with the AdamW optimizer using a learning rate of 1×10^{-5}. For fair comparison, Vanilla, Heuristic, Editor, and HCC use the same training configuration, and only differ in the SFT traces used for supervision. During evaluation, we fix the decoding temperature to 0 to ensure deterministic generation and improve reproducibility. All benchmark results are reported under this fixed evaluation setting.

A.2 Construction of Comparison Methods

Vanilla. We directly use the original answer-correct trajectories for SFT.

Editor. Qwen3.5-27B identifies a removable post-conclusion reasoning suffix under the delete-only protocol. We retain the preceding reasoning prefix and the original final answer.

Heuristic. Following Li et al. (2026), we compute the proportion of reasoning steps containing branching cues, such as "perhaps," "another," and "alternatively," and filter traces with a high branching-cue proportion.

HCC. HCC predicts a sentence-level cut boundary. We retain the reasoning prefix ending at the predicted boundary and preserve the original final answer.

B Information Bottleneck Motivation for Answer-Sufficiency

In this appendix, we provide an information-bottleneck motivation for the answer-sufficiency component of HCC. The goal is to formalize when a reasoning prefix can already support the final answer, and to give an idealized sufficient condition under which a suffix does not change the answer decision under an answer-prediction head. This formulation serves only as a conceptual motivation: the answer-prediction head and variational information-bottleneck objective introduced below are analytical devices, whereas the implemented HCC uses only the training objectives defined in Section 4.

B.1 Sequential Conditional Information Bottleneck

Given a question Q, a sentence-level reasoning trace $R = (r_1, \ldots, r_T)$, and the final answer A, let $R_{\le t}$ and $R_{>t}$ denote the prefix and suffix at step t. For each prefix $R_{\le t}$, we introduce a stochastic bottleneck representation Z_t induced from the contextualized reasoning state $\tilde{h_t}$:

$$q_\phi(z_t \mid \tilde{h_t}) = \mathcal{N}(\mu_t, \Sigma_t).$$

The ideal conditional information bottleneck objective aims to preserve answer-relevant information while removing unnecessary dependence on the reasoning prefix:

$$\begin{aligned} \min \quad & I(Z_t; R_{\le t} \mid Q) \\ \text{s.t.} \quad & I(Z_t; A \mid Q) \text{ is sufficiently large.} \end{aligned}$$

Equivalently, one may optimize the following Lagrangian form:

$$\mathcal{J}_{\mathrm{IB}} = -I(Z_t; A \mid Q) + \beta I(Z_t; R_{\le t} \mid Q).$$

Since exact mutual information is intractable for free-form reasoning traces, we use a variational approximation. The answer-relevance term is optimized through a variational answer predictor $p_\psi(a \mid z_t, q)$, leading to the negative log-likelihood term:

$$-\log p_\psi(a^* \mid z_t, q).$$

For the compression term, we introduce a sequential prior $p_\eta(z_t \mid z_{t-1}, q)$ and use the following KL penalty as a variational surrogate for the incremental information injected into the bottleneck:

$$D_{\mathrm{KL}}\big(q_\phi(z_t \mid \tilde{h}_t) \,\|\, p_\eta(z_t \mid z_{t-1}, q)\big).$$

Thus, the practical sequential IB loss is:

$$\mathcal{L}_{\mathrm{IB}} = \sum_{t=1}^{T} \Big[-\log p_\psi(a^* \mid z_t, q) + \beta D_{\mathrm{KL}}\big(q_\phi(z_t \mid \tilde{h}_t) \,\|\, p_\eta(z_t \mid z_{t-1}, q)\big)\Big].$$

This objective encourages Z_t to retain information predictive of the final answer while discouraging unnecessary state complexity.

B.2 Ideal Answer-Sufficiency Boundary

We next formalize an ideal boundary after which the remaining reasoning suffix provides no additional answer information under the bottleneck representation.

Definition. The ideal answer-sufficiency boundary is defined as:

$$\tau^* = \min\{t : I(A; R_{>t} \mid Q, Z_t) = 0\}.$$

This condition means that, once Q and Z_t are given, the suffix $R_{>t}$ provides no additional information about the final answer A. If the set is empty, the trace has no boundary satisfying this ideal condition.

Proposition. If

$$I(A; R_{>t} \mid Q, Z_t) = 0,$$

then conditioning on $R_{>t}$ does not change the Bayes-optimal answer distribution given (Q, Z_t).

Proof. By the definition of conditional mutual information, the condition

$$I(A; R_{>t} \mid Q, Z_t) = 0$$

implies the conditional independence relation:

$$A \perp R_{>t} \mid (Q, Z_t).$$

Therefore, for any answer candidate a, we have:

$$p(a \mid Q, Z_t, R_{>t}) = p(a \mid Q, Z_t).$$

As a result, the Bayes-optimal answer predictor satisfies:

$$\arg\max_a p(a \mid Q, Z_t, R_{>t}) = \arg\max_a p(a \mid Q, Z_t).$$

Thus, once Z_t is answer-sufficient in the conditional-independence sense, the suffix $R_{>t}$ is irrelevant to the answer decision under the bottleneck representation. □

Remark. This proposition establishes an answer-sufficiency condition only. It does not imply that deletion necessarily improves SFT. The empirical benefit studied in the main paper additionally depends on whether the answer-sufficient suffix introduces high local uncertainty or weak geometric progress as supervision.

C Experimental Settings

C.1 Uncertainty Metrics

We compute uncertainty metrics with a fixed evaluator model. Each answer-correct trace is split into sentence-level units, and the final boxed answer is used as the answer target. When scoring intermediate reasoning text, standalone boxed-answer strings are removed to reduce trivial answer leakage.

For sentence-level uncertainty, each sentence $r_t = (y_1, \ldots, y_m)$ is scored under its preceding context P_{t-1}. We report token-averaged NLL and predictive entropy:

$$\mathrm{NLL}_{\mathrm{sent}}(r_t) = -\frac{1}{m}\sum_{i=1}^{m} \log p(y_i \mid P_{t-1}, y_{<i}),$$

$$\mathrm{Ent}_{\mathrm{sent}}(r_t) = \frac{1}{m}\sum_{i=1}^{m} H(p(\cdot \mid P_{t-1}, y_{<i}))\,.$$

For answer-level uncertainty, we measure how a reasoning prefix affects recovery of the boxed answer $a^* = (a_1, \ldots, a_L)$. Given the prefix P_t after appending sentence r_t, we compute:

$$\mathrm{NLL}_{\mathrm{ans}}(P_t) = -\frac{1}{L}\sum_{i=1}^{L} \log p(a_i \mid P_t, a_{<i}).$$

Answer entropy is computed analogously over the answer-token positions. We define answer-NLL reduction as:

$$\Delta_{\mathrm{ans}}(t) = \mathrm{NLL}_{\mathrm{ans}}(P_{t-1}) - \mathrm{NLL}_{\mathrm{ans}}(P_t).$$

A larger value means that appending r_t makes the final answer easier to recover under the evaluator.

For segment-wise plots, sentences are appended along the original complete trace, while the x-axis is normalized separately within retained reasoning and editor-removed continuation. For boundary-level plots, K_1 and K_T denote the first and last retained sentences, and C_1 and C_T denote the first and last editor-removed sentences.

C.2 Geometric Metrics

We compute geometric metrics from hidden states at sentence boundaries. Let h_t denote the evaluator hidden state after consuming the prefix ending at sentence r_t. The local state update is:

$$\Delta h_t = h_t - h_{t-1}.$$

Hidden displacement measures the size of this update:

$$D_t = \|\Delta h_t\|_2.$$

Forward progress measures the projection of the local update onto the remaining direction toward the terminal state of the analyzed trace:

$$G_t = \frac{\langle \Delta h_t, h_T - h_{t-1} \rangle}{\|h_T - h_{t-1}\|_2 + \epsilon}.$$

This is an operational proxy for terminal-directional hidden-state progress, not a direct measurement of the true reasoning process.

Progress efficiency is defined as:

$$E_t = \frac{G_t}{D_t + \epsilon}.$$

To control for sentence length, we also report token-normalized variants:

$$D_t^{\mathrm{tok}} = \frac{D_t}{n_t}, \qquad G_t^{\mathrm{tok}} = \frac{G_t}{n_t},$$

where n_t is the token length of r_t.

Curvature is used only as an auxiliary direction-change diagnostic:

$$\mathrm{Curv}_t = 1 - \frac{\langle \Delta h_{t-1}, \Delta h_t \rangle}{\|\Delta h_{t-1}\|_2 \|\Delta h_t\|_2 + \epsilon}, \qquad t > 1.$$

For paired comparisons, we average each metric within retained reasoning and editor-removed continuation for each example. The paired difference is:

$$\Delta = \mathrm{Mean}_{\mathrm{removed}} - \mathrm{Mean}_{\mathrm{retained}}.$$

We report group means, the fraction of examples where the removed continuation is lower or higher, and the 95% confidence interval of Δ.

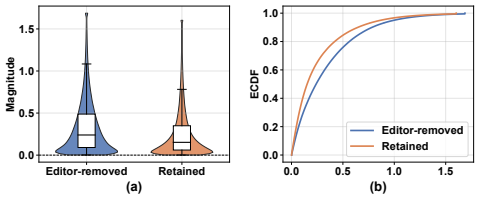


Figure 7: Additional uncertainty-side diagnosis. (a) NLL perturbation induced by retained reasoning and editor-removed continuation. (b) ECDF of average log-probability perturbation. Editor-removed continuation induces larger answer-level perturbations.

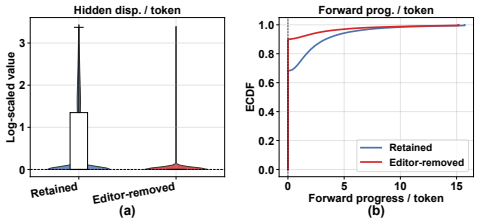


Figure 8: Additional geometry-side diagnosis. (a) Token-normalized hidden displacement of retained reasoning and editor-removed continuation. (b) ECDF of token-normalized forward progress. Editor-removed continuation is more concentrated in low-progress regions under the operational proxy.

D Additional Experiments

D.1 Additional Analysis of Harmful Continuation Diagnosis

Uncertainty View. Figure 7 compares the answer-level perturbation induced by retained reasoning and editor-removed continuation. Editor-removed continuation shows larger NLL perturbation and a right-shifted distribution of average log-probability perturbation. This suggests that the removed harmful continuation is not simply irrelevant text, but remains answer-conditioned and introduces stronger instability to evaluator-based final-answer prediction.

Geometric View. Figure 8 further compares the geometric behavior of the two groups. Retained reasoning induces larger token-normalized hidden displacement, while editor-removed continuation is more concentrated in the low forward-progress region. This is consistent with the view that editor-removed continuation corresponds to a low-progress phase: it can still affect answer prediction, but does not provide comparable representation-level state movement toward the terminal reasoning state. Together, these additional results support the uncertainty–geometry mismatch diagnosis of harmful continuation.

Table 5: Diagnostic robustness controls. Cross-editor results compare cut positions identified by `Qwen3.5-27B` and `DeepSeek-V4-Flash`.

Control	Metric	Result
Cross-editor	Normalized cut MAE ↓	0.047
	Lin's CCC ↑	0.734
	Exact match	51.65%
	Within 5%	78.61%
	Within 10%	87.48%
	Signed bias	0.0025
Position	Progress gap per token	−0.198

Table 6: Comparison of different methods across backbone models on the MMLU dataset. $\{T\}_Q$ and $\{T\}_R$ denote SFT trajectories from Qwen-style and R1-style long-CoT sources, respectively.

Method	LLaMA3.2-3B-Instruct		Qwen3-30B-A3B-Thinking-2507	
	$\{T\}_Q$	$\{T\}_R$	$\{T\}_Q$	$\{T\}_R$
Vanilla	60.1	58.0	<u>85.4</u>	84.6
Heuristic	59.8	60.1	85.0	85.3
Editor	61.1	60.6	85.2	<u>85.4</u>
HCC	62.9	60.8	**85.7**	**85.7**

Table 7: Pass@1 results of the Random-Fix and Random-Bin boundary controls using LLaMA3.2-3B-Instruct trained on $\{T\}_R$.

Method	MATH500	AMC23	GSM8K
Random-Fix	0.0	7.5	19.6
Random-Bin	11.4	12.5	18.9
HCC	**47.6**	**22.5**	**77.8**

D.2 Diagnostic Robustness Controls

Our diagnosis may be confounded by editor-specific boundary preferences or by the late position of the removed continuation. We examine these alternatives through cross-editor and within-response controls.

Agreement metrics. For trace i with T_i reasoning sentences, let c_i^Q and c_i^D denote the cut positions predicted by Qwen3.5-27B and DeepSeek-V4-Flash, respectively. We normalize each boundary by the trace length:

$$u_i^Q = \frac{c_i^Q}{T_i}, \qquad u_i^D = \frac{c_i^D}{T_i}. \tag{16}$$

The normalized cut-position MAE measures the average absolute boundary difference:

$$\mathrm{MAE}_{\mathrm{norm}} = \frac{1}{N}\sum_{i=1}^{N}\left|u_i^Q - u_i^D\right|. \tag{17}$$

It captures how far apart the two editors cut, irrespective of direction.

We use Lin's concordance correlation coefficient (CCC) (Lawrence & Lin, 1989) to measure whether the two editors identify boundaries with similar variation and similar absolute positions:

$$\mathrm{CCC} = \frac{2\,\mathrm{Cov}(u^Q, u^D)}{\mathrm{Var}(u^Q) + \mathrm{Var}(u^D) + (\mu_Q - \mu_D)^2}, \tag{18}$$

where μ_Q and μ_D are the mean normalized cut positions.

We additionally report the exact-match rate and the proportion of traces satisfying

$$\left|u_i^Q - u_i^D\right| \le \delta, \qquad \delta \in \{0.05, 0.10\}, \tag{19}$$

which distinguish exact sentence-level agreement from agreement within a local boundary neighborhood. Finally, signed bias is defined as

$$\text{Bias} = \frac{1}{N}\sum_{i=1}^{N}\left(u_i^D - u_i^Q\right), \tag{20}$$

where a positive or negative value indicates that the model tends to cut later or earlier, respectively.

Cross-editor consistency. To test whether the operational partition is specific to Qwen3.5-27B, we re-annotate the same traces with DeepSeek-V4-Flash under the same protocol. As shown in Table 5, the two editors obtain a normalized cut-position MAE of 0.047 and a Lin's CCC of 0.734. The small normalized error indicates that they generally identify nearby transition regions, while the concordance shows that boundary variation across traces is largely preserved between model families. Moreover, the confidence interval of the signed bias includes zero, providing no evidence that either editor systematically cuts earlier or later. The remaining disagreement is consistent with the absence of a unique sentence-level boundary: multiple nearby cuts may preserve the answer while removing similar continuation content. Thus, the operational partition captures a cross-editor transition pattern rather than the exact decisions of a single editor, although neither editor should be treated as a ground-truth boundary oracle.

Table 8: HCC-based self-consistency diagnostics. Phase, Sent. ratio, and Len. denote the occurrence rate of outputs matching the HCC removable-continuation pattern, the corresponding sentence-level ratio, and the average token length, respectively.

Dataset	Method	Phase ↓	Sent. ratio ↓	Len.
GSM8K	Vanilla	81.73	51.84	94.54
	Heuristic	93.63	59.91	96.19
	Editor	63.53	24.14	50.85
	HCC	**60.42**	**19.45**	**48.38**
MATH500	Vanilla	93.40	59.51	74.75
	Heuristic	93.60	60.06	73.99
	Editor	63.60	28.53	73.89
	HCC	**61.80**	**27.35**	**66.86**

Table 9: Comparison across reinforcement steps.

Method	Dataset	0	10	20	30	40
Vanilla	MATH500	29.8	34.8	35.2	36.6	36.4
	GSM8K	69.0	71.6	72.7	73.1	73.3
HCC	MATH500	43.2	45.8	46.0	46.2	49.4
	GSM8K	75.1	75.4	75.7	77.7	77.0

Table 10: Pass@1 results of Qwen3-30B-A3B-Instruct trained on original and HCC-processed mathematical long-CoT traces.

Training traces	AIME24	AIME25	MATH500	AMC23	GSM8K
Original	66.7	56.7	95.0	**92.5**	96.3
HCC-processed	**80.0**	**66.7**	**96.4**	87.5	**96.4**

Within-response positional control. To examine whether weaker geometric progress is explained solely by late position, we partition each response into retained reasoning, editor-removed continuation, and the final answer segment. The answer segment follows the complete reasoning trace and therefore occurs even later than the removed continuation. Using the metric defined in Appendix C.2, we compute:

$$\Delta_{\text{pos}} = \text{Mean}_{t\in C}\, G_t^{\text{tok}} - \text{Mean}_{t\in A}\, G_t^{\text{tok}}, \tag{21}$$

Table 11: Results of the non-math-specialized `Qwen2.5-7B` trained on original and HCC-processed trajectories. Pass@1 measures task accuracy, and Avg. Tok. denotes the average generated response length.

Training traces	MATH500		AMC23		GSM8K	
	Pass@1 ↑	**Avg. Tok. ↓**	**Pass@1 ↑**	**Avg. Tok. ↓**	**Pass@1 ↑**	**Avg. Tok. ↓**
Original	53.8	4745.21	35.0	6097.02	79.4	3287.24
HCC-processed	**62.2**	**2849.66**	**37.5**	**4834.55**	**85.4**	**1040.91**

Method	# Params	MACs	FLOPs	Rel. MACs
Qwen3.5-27B	27.0B	137.1T	274.3T	∼54.2×
HCC	498.0M	2.5T	5.1T	1.0×

Table 12: Computational cost comparison given similar input lengths of Qwen3.5-27B and HCC.

where C denotes the editor-removed continuation and A denotes the final answer segment.

The mean positional gap is -0.198, showing that the removed continuation has lower progress per token than the subsequent answer segment. If geometric progress decreased solely as a function of late position, the later answer segment would be expected to exhibit comparable or lower progress. The observed direction is inconsistent with this explanation, suggesting that late position alone does not account for the weak-progress pattern. Because the continuation and answer segments differ in content and function, this comparison should be interpreted as a within-response positional control rather than a causal estimate of the content effect. Together, these controls reduce two important diagnostic confounds. They do not independently establish that the removed continuation harms training; that conclusion is supported by the downstream answer-preserving deletion intervention reported in the main experiments.

D.3 Additional analysis of Test Datasets

We evaluate two randomized boundary controls using LLaMA3.2-3B-Instruct trained on $\{T\}_R$.

Construction of Randomized Boundary Controls. For Random-Fix, we independently sample a random sentence-boundary cut for each trace while constraining the resulting collection to have the same average retained token length as the HCC-processed collection. This control preserves the corpus-level training budget but does not use the content or state of an individual trace to determine its boundary. For Random-Bin, we collect the cut positions predicted by HCC and randomly permute their assignments across traces. The permutation preserves HCC's empirical cut-position distribution and overall amount of removed reasoning, while breaking the correspondence between a trace and its predicted boundary. Random-Bin therefore provides a stronger control than Random-Fix: it matches not only the overall amount of truncation but also where cuts tend to occur across the collection.

Analysis of Randomized Boundary Controls. As shown in Table 7, neither random control reproduces the downstream performance of HCC. Random-Fix shows that matching the average retained length alone is insufficient, while Random-Bin shows that preserving the global cut-position distribution is also insufficient when boundaries are assigned to the wrong traces. A randomly assigned boundary may fall too early and remove answer-supporting reasoning, or too late and retain the post-conclusion continuation. The contrast therefore supports the importance of conditioning the cut on each trace's reasoning trajectory, rather than treating shortening or global cut position as the source of HCC's benefit.

Downstream check for the cut-only variant. As an additional single-setting check, we use the cut-only boundary predictor to process $\{T\}_R$ and train LLaMA3.2-3B-Instruct. It obtains Pass@1 scores of 36.4, 15.5, and 70.6 on MATH500, AMC23, and GSM8K, respectively, below full HCC's 47.6, 22.5, and 77.8. This result provides supporting evidence that the auxiliary objectives improve

not only boundary agreement but also downstream trace utility, although the comparison is limited to one student and trajectory source.

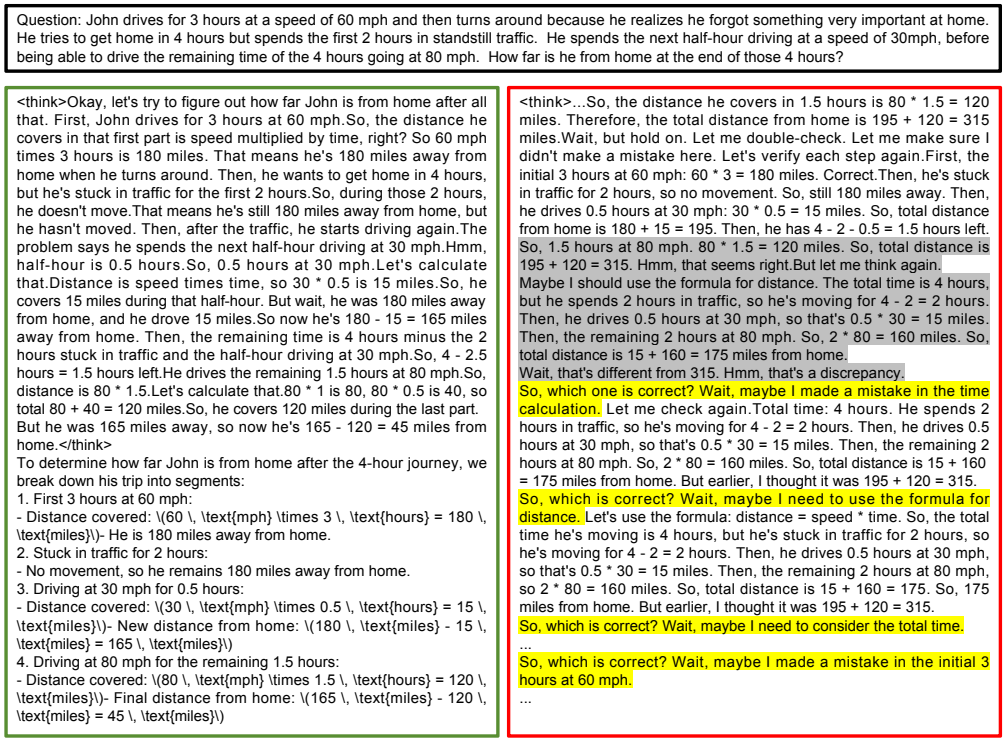


Figure 9: Case study of harmful continuation after SFT. The left part indicates the reasoning process of a model trained on HCC-processed traces, while the right part shows the reasoning process of a model trained on original traces. We use grey and yellow highlights to indicate the conflicting reasoning and the inefficient reasoning loop, respectively.

HCC-based self-consistency diagnostic. Table 8 evaluates whether model outputs after SFT match the removable post-conclusion continuation pattern learned by HCC. For a fair cross-source test, we use the HCC proxy trained on $\{T\}_Q$ to analyze outputs from models trained on $\{T\}_R$-based data. This analysis should be interpreted as a detector-based self-consistency diagnostic rather than an independent measurement of causal harmfulness. Under this diagnostic, models trained on HCC-processed traces produce fewer outputs that are classified as containing removable post-conclusion continuation. For example, HCC reduces the sentence-level detected continuation ratio from 51.84% to 19.45% on GSM8K and from 59.51% to 27.35% on MATH500.

Persistence through subsequent GRPO. We examine whether the performance difference established during SFT persists after subsequent GRPO in one controlled setting. Starting from LLaMA3.2-3B-Instruct checkpoints trained on original or HCC-processed $\{T\}_Q$ traces, we apply the same GRPO configuration and evaluate both models at matched training steps. As shown in Table 9, the model initialized from HCC-based SFT maintains higher performance at every evaluated step. This result indicates that, in this setting, the advantage associated with HCC-processed SFT is retained rather than immediately erased by subsequent GRPO; it does not by itself establish the same behavior across other students, trajectory sources, or RL algorithms.

D.4 Evaluation on a Larger Student

To examine whether HCC-processed traces also benefit a substantially larger student model, we evaluate Qwen3-30B-A3B-Instruct (Team, 2025). We fine-tune the model on either the original or HCC-processed mathematical long-CoT traces using $\{T\}_R$. Because this strong student already

obtains high scores on several standard benchmarks, we additionally evaluate it on the more challenging AIME24 and AIME25 benchmarks. As shown in Table 10, HCC-processed traces increase Pass@1 from 66.7 to 80.0 on AIME24 and from 56.7 to 66.7 on AIME25. The two settings perform similarly on GSM8K and MATH500, where the larger student already achieves high accuracy, while HCC is lower on AMC23 (87.5 versus 92.5). These results indicate that HCC-processed traces can benefit a larger student, particularly on the more challenging AIME benchmarks, but do not produce uniform improvements across all evaluations.

D.5 Effect of Prior Math-Specific SFT

The main results show weaker and less consistent gains from HCC on Qwen2.5-Math-7B. One possible explanation is that this model has already undergone SFT on high-quality mathematical data and may therefore be less sensitive to additional trace processing. To examine this explanation, we evaluate the non-math-specialized `Qwen2.5-7B`, which has not undergone prior math-specific SFT. We fine-tune it on either the original or HCC-processed trajectories using $\{T\}_R$. As shown in Table 11, HCC-processed traces improve Pass@1 on all three benchmarks while also reducing average response length. This contrasts with the weaker results on Qwen2.5-Math-7B and is consistent with the interpretation that prior math-specific SFT reduces the model's sensitivity to additional trace processing.

Case study. Figure 9 presents a qualitative example comparing the same base model trained with HCC-processed traces and original long-CoT traces. The HCC-trained model quickly identifies the correct reasoning path, computes the remaining distance, and outputs the correct solution without entering a long verification loop. In contrast, the Vanilla-trained model first reaches the correct answer, but then derives a conflicting result from an alternative calculation, as shown in the gray region. After this conflict appears, the model repeatedly compares the two answers and rechecks different parts of the solution, entering a low-efficiency reasoning loop highlighted in yellow. The response eventually exhausts the token budget without producing a successful final answer. This example qualitatively illustrates a behavior consistent with the harmful continuation pattern: the model may continue uncertain, low-progress verification even after a sufficient answer has already been reached. It suggests that HCC-processed supervision can reduce such continuation patterns.

Analysis of Computational Costs. Table 12 compares the computational cost of the 27B offline editor and our HCC proxy given similar input lengths. HCC uses only 498M parameters, which is about 1.8% of Qwen3.5-27B. It also requires 2.5T MACs and 5.1T FLOPs, compared with 137.1T MACs and 274.3T FLOPs for Qwen3.5-27B. This corresponds to a roughly $54.2\times$ reduction in computation. These results show that HCC provides a much cheaper proxy for editor-identified harmful continuation boundary approximation, making large-scale SFT trace processing more practical.