



Confidence-Aware Reasoning: Optimizing Self-Guided Thinking Trajectories in Large Reasoning Models

Jiaxin Zhang

Independent Researcher

Abstract

Chain-of-thought enables large reasoning models (LRMs) to reason through multi-step problems but often leads to unnecessarily long or redundant reasoning traces, a phenomenon known as *overthinking*. This results in inflated inference costs and potential degradation in answer quality. To address these challenges, we propose Confidence-Aware Reasoning (CaR), an inference-time framework that optimizes reasoning trajectories by selectively pruning low-utility reasoning blocks and halting early when sufficient confidence has been achieved. CaR is theoretically grounded in Bayesian optimal experimental design, treating each reasoning block as a sequential decision whose utility is approximated by its marginal contribution to reducing final answer uncertainty. We introduce a lightweight implementation that leverages token-level confidence to dynamically modulate reasoning depth without additional supervision. Evaluations on multiple benchmarks, including AMC, AIME, GPQA-Diamond, and MATH-500 show that CaR improves answer accuracy by up to +13.3%, while reducing average reasoning length by 40%–50%. Our findings demonstrate that information-theoretic insights can effectively control self-guided reasoning and enable LRMs to “think just enough” at test time.

1 Introduction

Large language models (LLMs) have achieved remarkable progress in multi-step reasoning through the use of chain-of-thought (CoT) prompting, where models are guided to first generate an intermediate sequence of thoughts before producing a final answer (Wei et al., 2023). Recent open-source models such as DeepSeek-R1 (DeepSeek-AI, 2025) and OpenAI’s o-series (Jaech et al., 2024) have further adopted System-2-style generation (Li et al., 2025), producing explicit reasoning traces that mirror deliberative thinking. These advances have led

to strong performance on complex benchmarks. However, this improvement often comes at the cost of significantly increased inference latency, as long reasoning trajectories are generated regardless of their necessity or informativeness (Wu et al., 2025; Cuadron et al., 2025).

In practice, many CoT traces include redundant or low-value steps that contribute little to final answer quality, sometimes even reducing it, a phenomenon known as *overthinking* (Chen et al., 2024; Su et al., 2025; Fan et al., 2025). This leads to both computational inefficiency and potential reasoning degradation, especially in settings with hard latency or token budgets (Muennighoff et al., 2025). Existing works have observed that a substantial fraction of reasoning tokens can be safely omitted without impacting the final answer (Ma et al., 2025), motivating the need for adaptive reasoning control. Prior efforts to address this issue fall into three main categories (Xu et al., 2025a): (1) post-training approaches that fine-tune the model using variable-length CoT or reward shaping (Luo et al., 2025b; Munkhbat et al., 2025), (2) prompt-based heuristics that instruct the model to “be concise” (Aytes et al., 2025; Xu et al., 2025b), and (3) inference-time methods that monitor generated traces and apply early exit strategies (Yang et al., 2025; Zhang et al., 2025; Yi and Wang, 2025). Among these, inference-time approaches are appealing due to their plug-and-play nature, but they typically rely on either external probes or ad hoc thresholds, and lack a principled framework for deciding when to prune or terminate reasoning.

We address these limitations by proposing a novel inference-time framework **Confidence-aware Reasoning (CaR)**, which optimizes self-guided reasoning trajectories through information-theoretic signals. At the heart of our approach lies a simple yet powerful insight: each reasoning step should be evaluated not merely for fluency, but for its *potential to reduce uncertainty* about

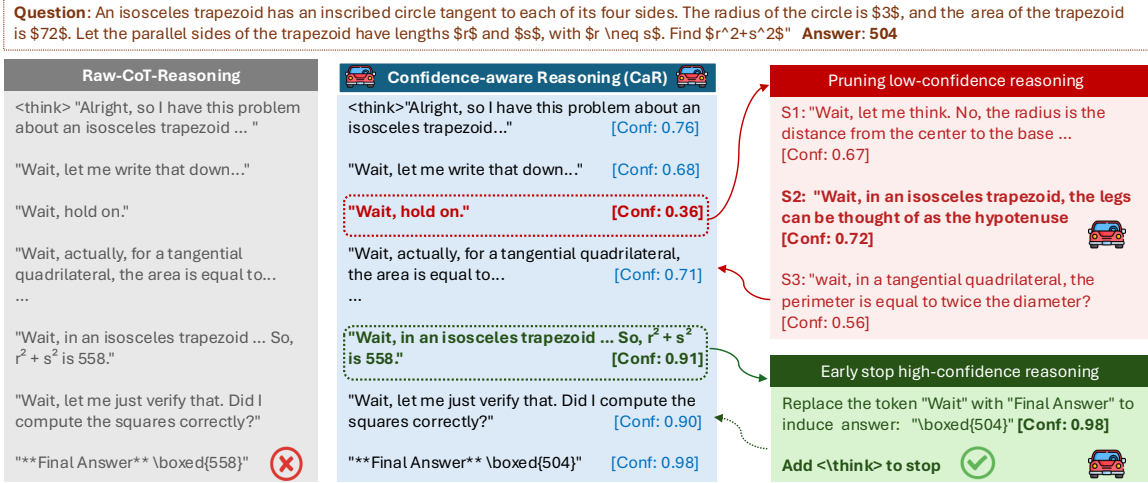


Figure 1: Overview of Confidence-Aware Reasoning (CaR). An illustrative example of CaR applied to a mathematical problem. The model generates intermediate reasoning blocks (separated by “Wait”), each associated with a token-level confidence score. Low-confidence steps are pruned, while high-confidence blocks are retained. When the confidence for both the reasoning and the candidate answer exceeds predefined thresholds, CaR inserts a `<think>` token to stop further generation and outputs the final answer. This process adaptively controls reasoning depth and improves efficiency without modifying model parameters. See Section 3 for more details.

the final answer. Inspired by Bayesian optimal experimental design, we interpret each intermediate reasoning block as a design decision whose utility can be approximated by the LRM’s predictive confidence. This intuition enables two complementary interventions: **pruning** low-confidence blocks that likely contribute little information gain, and **early stopping** when the model exhibits strong internal certainty. Unlike existing early-exit or probe-based strategies (Yang et al., 2025; Zhang et al., 2025), CaR operates entirely within the self-guided decoding loop, requiring no fine-tuning or external feedback. CaR dynamically modulates reasoning length based on token-level confidence, allowing the model to “think just enough”—longer when uncertain, shorter when confident. An illustrative example is provided in Figure 1.

2 Preliminaries

2.1 Problem Formulation

Let P denote a problem statement that necessitates complex reasoning, such as math or programming tasks. We employ a reasoning language model $\mathcal{L}$ and the inference begins with the construction of an initial prompt $\mathcal{P}$, which instructs $\mathcal{L}$ to generate step-by-step reasoning enclosed within predefined delimiters (e.g., “<think>...</think>”), followed by a final answer presented in a standardized format (e.g., “\boxed{Answer}”). Given the problem prompt $\mathcal{P}$, we call the LRM $\mathcal{L}$ to generate

the full output response $\mathcal{O}$ via inference decoding:

$$\mathcal{O} = \mathcal{L}(\mathcal{P} \mid \theta), \quad (\mathcal{R}, \mathcal{A}) \in \mathcal{O}, \quad (1)$$

where θ is the parameter of LRMs, and the output $\mathcal{O}$ consists of the thinking/reasoning process $\mathcal{R}$ and the final answer $\mathcal{A}$. Specifically, the thinking process can typically be divided into multiple reasoning blocks $\mathcal{R} = \{r_1, \dots, r_n\}$.

Our objective is to *maximize the performance of the final answer by self-guiding the thinking trajectories while avoiding overthinking*. Formally, we define the optimization problem as:

$$\max_{\mathcal{R}} \text{Perf}(\mathcal{A}) \quad \text{subject to} \quad \text{Len}(\mathcal{R}) \leq \tau, \quad (2)$$

where $\text{Perf}(\mathcal{A}) \in [0, 1]$ denotes a task-specific performance metric of the final answer $\mathcal{A}$, $\text{Len}(\mathcal{R})$ denotes the length of the reasoning trace $\mathcal{R}$, and τ is a predefined threshold controlling the maximal allowable reasoning length.

2.2 Bayesian Optimal Experimental Design

We draw theoretical motivation from the framework of Bayesian Optimal experimental design (BOED) (Foster et al., 2019), which formalizes how to make optimal decisions under uncertainty by acquiring the most informative observations (Chaloner and Verdinelli, 1995). BOED provides a general principle for **sequential information acquisition** which inspires us to reinterpret this perspective by treating each reasoning block as a design choice, and

the eventual answer as the latent target variable. This analogy allows us to approximate the marginal utility of each block based on how much it is expected to reduce uncertainty over the final answer distribution. As a result, BOED offers a natural foundation for our CaR framework, guiding both the **pruning** and **stopping** of reasoning steps based on information-theoretic utility. More discussions are provided in the appendix.

3 Methodology

3.1 Information-Theoretic Insights

We reinterpret the principle of BOED by viewing each intermediate **reasoning block** r_t as a local design decision ξ_t , whose purpose is to reduce the reasoning model’s epistemic uncertainty over the correct final answer $\mathcal{A}$. Let $\mathcal{R}_{<t} = \{r_1, r_2, \dots, r_{t-1}\}$ denote the partial reasoning trajectory up to step $t - 1$. The final output consists of a full trajectory $\mathcal{R} = \mathcal{R}_{<t} \cup \{r_t\}$ and an answer $\mathcal{A}$. We define the marginal utility of block r_t as the conditional EIG over the answer distribution:

$$\text{EIG}_{\mathcal{A}} = \mathbb{E}_{\mathcal{A} \sim p(\cdot | \mathcal{R}_{<t}, r_t)} \left[\log \frac{p(\mathcal{A} | \mathcal{R}_{<t}, r_t)}{p(\mathcal{A} | \mathcal{R}_{<t})} \right].$$

This quantity quantifies the expected information contribution of r_t toward the final answer $\mathcal{A}$. Our objective is to select the next reasoning block r_t that maximizes the conditional EIG with respect to the final answer $\mathcal{A}$: $r_t^* = \arg \max_{r_t} \{\text{EIG}_{\mathcal{A}}\}$. However, directly estimating the $\text{EIG}_{\mathcal{A}}$ is intractable in practice, as it requires full posterior access or Monte Carlo sampling over alternative reasoning paths, which is computationally expensive for LLMs (Rainforth et al., 2024).

To overcome this limitation, we propose a tractable **single-sample proxy** for the $\text{EIG}_{\mathcal{A}}$ of each reasoning block, leveraging the model’s token-level predictive confidence. This proxy avoids Monte Carlo sampling, making it tractable for large-scale decoding. Let $r_t = \{y_1^{(t)}, \dots, y_k^{(t)}\}$ be the tokens comprising the t -th block, $\mathcal{C}$ means the entire previous “context”, and define the block confidence $\text{Conf}(r_t)$ as the average log-probability of its tokens.

Empirical Observations. We note that blocks with low predictive confidence tend to be semantically vague or uninformative (e.g., filler phrases such as “Wait”, “Wait, no”, “Wait, but...”). Figure 3 visualizes the manifold of the low-confidence

reasoning blocks (e.g., $\text{Conf}(r_t) \leq 0.6$). We can observe isolated clusters from both sizes of models (1.5B and 14B), and the zoomed-in visualization with annotated tokens are shown in Figure 2.

However, high-confidence blocks often correlate with decisive and informative steps in the reasoning trajectory. Therefore, we approximate the $\text{EIG}_{\mathcal{A}}$ of r_t via a monotonic mapping of its confidence score: $\text{EIG}_{\mathcal{A}}(r_t | \mathcal{R}_{<t}) \approx \varphi(\text{Conf}(r_t))$, where $\varphi(\cdot)$ is an empirically calibrated function (e.g., identity, exponential, or entropy-based mapping) that serves as a surrogate for expected utility.

3.2 Pruning Low-Confidence Reasoning Blocks

Motivated by the information-theoretic insights and empirical observations, we propose a pruning mechanism to remove low-confidence reasoning blocks during inference, thereby reducing noise, instability, and risk of overthinking. We define a confidence threshold $\tau_{\text{low}} \in (0, 0.6)$. If $\text{Conf}(r_t) < \tau_{\text{low}}$, the block is considered **low-confidence**, and thus unlikely to yield nontrivial information gain:

$$\text{Conf}(r_t) < \tau_{\text{low}} \implies \widehat{\text{EIG}}_{\mathcal{A}}(r_t | \mathcal{R}_{<t}) \approx 0. \quad (3)$$

In this case, we prune r_t and return to the previous reasoning state $\mathcal{R}_{<t}$, from which we perform resampling. Specifically, we adopt the following pruning and resampling procedure:

- **Detection:** If $\text{Conf}(r_t) < \tau_{\text{low}}$, discard the current block r_t due to low IG.
- **Resampling:** Revert to the previous state $\mathcal{R}_{<t}$, and resample K candidate reasoning blocks with a relatively higher temperature:

$$\{r_t^{(1)}, \dots, r_t^{(K)}\} \sim \text{Decode}(\mathcal{L}(\mathcal{R}_{<t}, \theta)), \quad (4)$$

- **Selection:** Among the K candidates, select the block with the highest confidence, i.e., maximum information gain (IG):

$$r_t^* = \arg \max_{r_t^{(i)}} \text{Conf}(r_t^{(i)}), \quad i = 1, \dots, K. \quad (5)$$

- **Acceptance or fallback:** If $\text{Conf}(r_t^*) \geq \tau_{\text{low}}$, accept r_t^* as the new reasoning block; otherwise, accept it anyway to avoid infinite resampling, and proceed to step $t + 1$.

This mechanism can be viewed as a light-weight form of importance/reject sampling, which efficiently mitigates the inclusion of noisy reasoning fragments. Intuitively, it also improves alignment

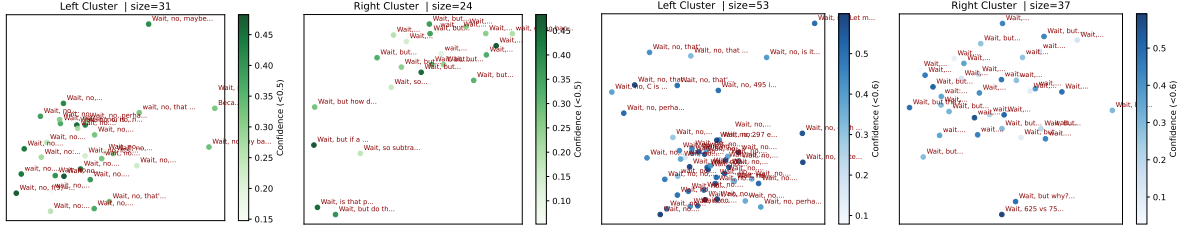


Figure 2: Zoom-in visualization of low-confidence tokens in low-dimensional space on 1.5B (green) and 14B (blue).

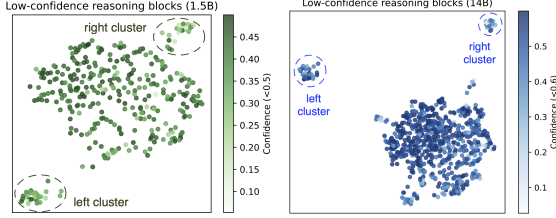


Figure 3: Visualization of low-confidence reasoning blocks on 1.4B and 15B models across AIME tasks.

between the LRM’s output space and its internal belief: by preferring blocks with higher confidence, we approximate the objective of maximizing IG with respect to the final answer distribution.

3.3 Early Stopping via High-Confidence Control

We further introduce an early stopping mechanism that terminates the reasoning trajectory once sufficient evidence has been accumulated. To avoid premature termination, we design a two-stage gating procedure with progressively stricter confidence thresholds. CaR first checks the high-confidence reasoning block, and then passes a second confidence check on the predicted final answer before halting. This approach ensures that the decision to stop is both context-aware and answer-aware.

Stage 1: Confidence Gate on the Reasoning Block. Let r_t be the current reasoning block, and define its normalized confidence score:

$$\text{Conf}(r_t) = \exp \left(\frac{1}{|r_t|} \sum_{j=1}^{|r_t|} \log p_{\theta}(y_j^{(t)} | \mathcal{C}_{j-1}) \right).$$

We define a threshold $\tau_{\text{high}} \in (0.7, 0.9)$. If $\text{Conf}(r_t) \geq \tau_{\text{high}}$, we hypothesize that the model has reached a confident and informative state in its trajectory. We initiate a probe to assess whether the model is ready to produce the final answer.

Stage 2: Confidence Gate on the Final Answer Conditioned on the current reasoning trace, we prompt the LRM $\mathcal{L}$ to generate a candidate answer

sequence $\mathcal{A}_t = \{a_1, \dots, a_{|\mathcal{A}_t|}\}$. We compute the answer confidence as:

$$\text{Conf}(\mathcal{A}_t) = \exp \left(\frac{1}{|\mathcal{A}_t|} \sum_{j=1}^{|\mathcal{A}_t|} \log p_{\theta}(a_j | \mathcal{R}, a_{<j}) \right)$$

If the answer confidence exceeds a stricter threshold $\tau_{\text{ans}} \in (0.9, 1.0)$, we consider the reasoning process complete and terminate the trajectory:

$$\text{Conf}(\mathcal{A}_t) \geq \tau_{\text{ans}} \implies \text{Emit } \mathcal{A}_t \text{ and stop.}$$

We then insert a special delimiter token (e.g., “<think>”) to signal the end of the reasoning phase and proceed to output the final answer. Otherwise, if the answer confidence is insufficient, we discard the candidate answer and resume the generation of the next reasoning block r_{t+1} .

Information-Theoretic Interpretation. This two-stage mechanism aligns with the goal of information-efficient reasoning. Stage 1 identifies the onset of marginal utility saturation, i.e., when the expected information gain of an additional reasoning block becomes negligible:

$$\widehat{\text{EIG}}_{\mathcal{A}}(r_{t+1} | \mathcal{R}_t) \approx 0. \quad (6)$$

Stage 2 provides a formal confirmation that the model’s predictive posterior over the answer $\mathcal{A}$ is sufficiently peaked:

$$H[p(\mathcal{A} | \mathcal{R}_t)] \leq \varepsilon \Leftrightarrow \text{Conf}(\mathcal{A}_t) \geq \tau_{\text{ans}}, \quad (7)$$

where $H[\cdot]$ denotes Shannon entropy and ε is a user-defined entropy budget. Thus, early stopping occurs only when both: (1) the latest reasoning step is confident enough to warrant a prediction attempt. (2) The answer itself is deemed reliable given the current trace. This two-stage design avoids overthinking and redundant reasoning, while maintaining rigorous guarantees on prediction quality.

A step-by-step workflow is provided in Algorithm 1. Unlike fixed-length generation or step-based heuristics, our CaR method is fully adaptive, self-guided, driven by the LRM’s own confidence over the reasoning blocks and the final answer.

Model	Method	MATH-500	AMC-2023	GPQA-Diamond	AIME-2023	AIME-2024	AIME-2025
Distill-Qwen-1.5B	CoT-Raw	65.4	42.5	5.7	20.0	20.0	16.7
	CaR	74.2 ($\uparrow 8.8$)	55.0 ($\uparrow 12.5$)	6.6 ($\uparrow 0.9$)	20.0 ($\uparrow 0.0$)	23.3 ($\uparrow 6.6$)	20.0 ($\uparrow 3.3$)
Distill-Qwen-7B	CoT-Raw	86.0	75.0	24.8	40.0	36.7	26.7
	CaR	87.4 ($\uparrow 1.4$)	85.0 ($\uparrow 15.0$)	28.3 ($\uparrow 3.5$)	50.0 ($\uparrow 10.0$)	46.7 ($\uparrow 10.0$)	33.3 ($\uparrow 6.6$)
Distill-Qwen-14B	CoT-Raw	86.0	82.5	52.0	73.3	66.7	40.0
	CaR	88.2 ($\uparrow 2.2$)	90.0 ($\uparrow 7.5$)	59.1 ($\uparrow 7.1$)	80.0 ($\uparrow 6.7$)	76.7 ($\uparrow 10.0$)	53.3 ($\uparrow 13.3$)
QwQ-32B	CoT-Raw	86.4	87.5	47.5	69.7	56.7	56.7
	CaR	87.8 ($\uparrow 1.4$)	85.0 ($\downarrow 2.5$)	52.5 ($\uparrow 5.0$)	69.7 ($\uparrow 0.0$)	70.0 ($\uparrow 13.3$)	53.3 ($\downarrow 3.4$)

Table 1: Accuracy results on the DeepSeek-R1-Distill series of models (1.5B, 7B, and 14B) and QwQ-32B. Accuracy improvements are shown in blue with $\uparrow$; regressions are shown in red with $\downarrow$.

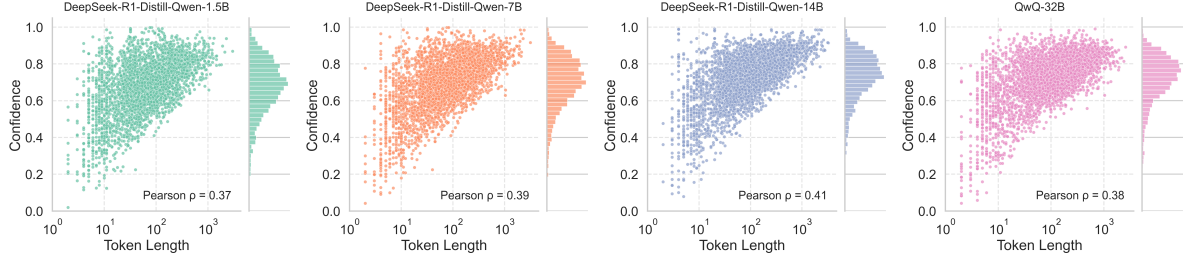


Figure 4: Relationship between the confidence of reasoning blocks and their token length across different LRMs.

Algorithm 1 Confidence-Aware Reasoning

```

1: Requirements: problem prompt  $\mathcal{P}$ ,  $\mathcal{P}_A$ , LRM  $\mathcal{L}_\theta$ , thresholds  $\tau_{\text{low}}$ ,  $\tau_{\text{high}}$ ,  $\tau_{\text{ans}}$ , budget  $\tau_{\text{budget}}$ , resample trials  $K$ 
2: Initialization: reasoning trajectory  $\mathcal{R} \leftarrow \emptyset$ , reasoning token length  $T \leftarrow 0$ 
3: while  $T \leq \tau_{\text{budget}}$  do
4:   Generate reasoning block  $r_t \sim \mathcal{L}_\theta(\mathcal{P} \mid \mathcal{R})$ 
5:   Compute  $\text{Conf}(r_t)$ 
6:   if  $\text{Conf}(r_t) < \tau_{\text{low}}$  then // Reasoning Pruning
7:     Re-sample  $K$  reasoning blocks  $r_t^{(k)}$ 
8:     Compute  $\text{Conf}(r_t^{(k)})$  for each sample
9:     Select  $r_t^* \leftarrow \arg \max_k \text{Conf}(r_t^{(k)})$ 
10:  end if
11:  if  $\text{Conf}(r_t) > \tau_{\text{high}}$  then // Early Stopping
12:    Generate candidate answer  $\mathcal{A}_t \sim \mathcal{L}_\theta(\mathcal{P}, \mathcal{P}_A \mid \mathcal{R})$ 
13:    Compute  $\text{Conf}(\mathcal{A}_t)$ 
14:    if  $\text{Conf}(\mathcal{A}_t) \geq \tau_{\text{ans}}$  then
15:      Append  $\langle \text{think} \rangle$  to  $\mathcal{R}$  and exit thinking
16:    end if
17:  end if
18:  Append  $r_t$  to  $\mathcal{R}$  and update  $T \leftarrow T + \text{Len}(r_t)$ 
19: Generate final answer  $\mathcal{A} \sim \mathcal{L}_\theta(\cdot \mid \mathcal{R})$ 
20: return  $\mathcal{A}, T$ 

```

4 Experiments

4.1 Experimental Setup

We evaluate our confidence-aware reasoning CaR framework on a diverse set of LRMs, including the DeepSeek-R1-Distill-Qwen family (**1.5B**, **7B**, **14B**) (DeepSeek-AI, 2025) and the **QwQ-32B** (Team, 2025) model. We select 6 benchmark datasets, including **AMC 2023**, **AIME 2023, 2024, 2025**, **MATH-500** (Hendrycks et al., 2021), and **GPQA-Diamond** (Rein et al., 2024) that emphasize com-

plex multi-step math reasoning. We follow prior work in adopting a zero-shot chain-of-thought prompting setup, and we report exact match accuracy on the final answers. Our evaluation focuses on two complementary objectives: reasoning accuracy, and computational efficiency. We aim to maximize the quality of the final answer while minimizing the length of the reasoning trajectory. More details are provided in the appendix.

4.2 Main Results

Reasoning Accuracy. As shown in Table 1-2, CaR consistently improves accuracy while substantially reducing the number of tokens generated during the reasoning process. This demonstrates that inference-time trajectory optimization can yield both performance and efficiency gains without tuning model parameters. Specifically, CaR improves accuracy by +12.5 on AMC-2023 and +8.8 on MATH-500 on the 1.5B model. Gains remain substantial at 7B and 14B scales, with improvements of +15.0 and +7.5, respectively. On challenging datasets like GPQA-Diamond and AIME-2025, CaR improves performance by up to +13.3.

Reasoning Length. CaR delivers significant reductions in reasoning token length, typically by 40–50%. For instance, average token length drops from 4205 to 2283 on AIME-2024 (−46%), and from 3864 to 2008 on GPQA-Diamond (−48%) on the 7B model. Even on the QwQ-32B model, which is already highly optimized for long-range

Model	Method	MATH-500	AMC-2023	GPQA-Diamond	AIME-2023	AIME-2024	AIME-2025
Distill-Qwen-1.5B	CoT-Raw	2053	3295	4387	4355	4601	5472
	CaR	1204 ($\downarrow 41\%$)	1585 ($\downarrow 52\%$)	2487 ($\downarrow 43\%$)	2068 ($\downarrow 53\%$)	2587 ($\downarrow 44\%$)	2749 ($\downarrow 50\%$)
Distill-Qwen-7B	CoT-Raw	1874	2658	3864	3901	4205	4954
	CaR	993 ($\downarrow 47\%$)	1539 ($\downarrow 42\%$)	2008 ($\downarrow 48\%$)	2177 ($\downarrow 44\%$)	2283 ($\downarrow 46\%$)	2558 ($\downarrow 48\%$)
Distill-Qwen-14B	CoT-Raw	1905	2866	4013	4823	4799	5293
	CaR	1055 ($\downarrow 45\%$)	1532 ($\downarrow 47\%$)	1955 ($\downarrow 51\%$)	2394 ($\downarrow 50\%$)	2693 ($\downarrow 44\%$)	2834 ($\downarrow 46\%$)
QwQ-32B	CoT-Raw	1977	3019	4482	5064	5385	6077
	CaR	1202 ($\downarrow 39\%$)	1699 ($\downarrow 44\%$)	2743 ($\downarrow 39\%$)	2793 ($\downarrow 45\%$)	3058 ($\downarrow 43\%$)	3864 ($\downarrow 36\%$)

Table 2: Token length results on the DeepSeek-R1-Distill series of models (1.5B, 7B and 14B) and QwQ-32B. Red $\downarrow$ indicates the reduction percentage of CaR compared to CoT-Raw .

reasoning, CaR reduces CoT length by 36–45% without degrading accuracy on most benchmarks.

These results underscore the core advantage of CaR: **generating less but thinking better**. By pruning low-utility reasoning blocks and terminating early when confidence peaks, the model avoids redundant or misleading paths while maintaining or improving prediction quality. Importantly, CaR operates entirely at the decoding level, making it broadly compatible with existing LRMs without requiring retraining or external supervision.

4.3 Analysis

Impact of confidence thresholds. We begin our ablation study by analyzing the effect of the two core hyperparameters in CaR : the low-confidence pruning threshold τ_{low} and the high-confidence early-stopping threshold τ_{high} .

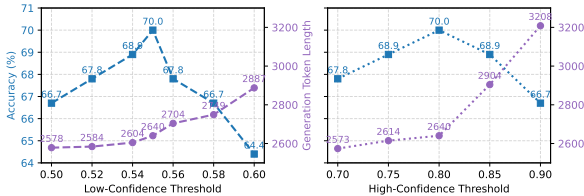


Figure 5: Effect of low/high-confidence thresholds on accuracy and generation token length.

As shown in Figure 5 (left), varying τ_{low} from 0.50 to 0.60 reveals a non-monotonic relationship with final accuracy. Moderate pruning (e.g., $\tau_{\text{low}} = 0.55$) yields the best accuracy (70.0%), while more aggressive pruning ($\tau_{\text{low}} \geq 0.58$) leads to a decline in performance. This suggests that overly strict pruning risks discarding useful intermediate steps, while too lenient thresholds retain noisy reasoning blocks. We therefore adopt $\tau_{\text{low}} = 0.55$ as a balanced default.

For τ_{high} , which determines when to attempt early answer generation, we observe in Figure 5 (right) that setting this value to 0.80 achieves

the best trade-off between accuracy and efficiency. As τ_{high} increases beyond 0.85, accuracy begins to drop due to premature answer emission, while smaller values (e.g., 0.70) incur higher token costs without substantial gains. We select $\tau_{\text{high}} = 0.80$ as the default in all experiments. Figure 4 shows the relationship between the confidence of reasoning blocks and their corresponding token length, which also provides some insights into the choice of confidence thresholds.

We also set an answer-level confidence threshold τ_{ans} for the final stopping condition. Prior work (Yang et al., 2025) observed that lower values (e.g., 0.90) often lead to premature stopping with under-developed CoT traces, while values close to 1.0 delay exit unnecessarily. The choice of 0.95 strikes a strong empirical balance between early termination and answer certainty, and avoids introducing instability in confidence estimation. We adopt $\tau_{\text{ans}} = 0.95$ throughout our experiments without additional tuning.

Block Pruning vs. Early Stopping. To disentangle the individual contributions of block pruning and early stopping (ES) within CaR , we conduct a component-wise ablation study across four models, reporting both final answer accuracy and average reasoning token length in Figure 6.

Introducing pruning consistently improves accuracy while substantially reducing token usage over the CoT-Raw baseline. For example, pruning raises accuracy from 34.5% to 38.9% and reduces generation length from 4353 to 2339 tokens (−46%) on 7B model. Similar trends are observed for 1.5B and 32B models, indicating that pruning low-confidence blocks helps remove distractive or redundant steps in early-stage reasoning.

Adding early stopping on top of pruning yields further efficiency gains, and in most cases, additional accuracy improvement. Notably, on 1.5B model, accuracy rises from 20.0% (CoT-Raw) to

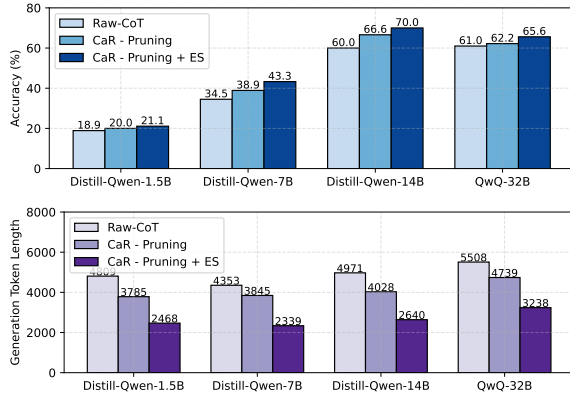


Figure 6: Effect of pruning and early stopping.

21.1% (Pruning + ES), while token length drops from 4809 to 2468 (−49%). On 14B model, however, we observe a small drop in accuracy when ES is applied (from 70.0% to 65.6%), suggesting that early halting may occasionally truncate valid yet long reasoning chains in high-capacity models.

5 Industrial Application

Our CaR approach has broad potential for real-world industrial applications where efficiency and reliability are critical. For example, CaR can reduce computation cost and latency for customer service chatbots, search and retrieval systems, and code assistants by pruning unnecessary reasoning steps while maintaining or even improving answer accuracy. It can also be used in edge or mobile deployments to fit strict token or memory budgets without requiring any model fine-tuning. Overall, CaR offers a simple yet effective method to optimize LRMs for production systems that need to balance accuracy, speed, and cost.

6 Related Work

Overthinking and Efficient Reasoning. Recent advances in LRMs, like Open-AI o1 (Jaech et al., 2024), and DeepSeek-R1 (DeepSeek-AI et al., 2025) have demonstrated that explicit chain-of-thought (CoT) generation can significantly improve performance across math, science, and logic benchmarks (Xu et al., 2025a; Li et al., 2025). However, this improvement often comes at the cost of increased test-time compute, as multi-step reasoning requires longer generation traces compared to standard prompting (Guo et al., 2025; Feng et al., 2025; Muennighoff et al., 2025). Moreover, these models frequently engage in unnecessary or repetitive reasoning steps—commonly referred to as

“overthinking”, even after the correct answer has been implicitly reached (Chen et al., 2024; Zhang et al., 2025). To mitigate this, several methods have been proposed to improve reasoning efficiency, including training-time interventions that encourage concise reasoning (Munkhbat et al., 2025), and test-time strategies that adaptively constrain generation based on confidence signals, prompt complexity, or answerability heuristics (Zhao et al., 2024; Manvi et al., 2024; Li et al., 2024; Yang et al., 2025; Xu et al., 2025b; Ma et al., 2025; Luo et al., 2025a; Hammoud et al., 2025).

Information-Theoretic Perspective. A principled view of efficient reasoning emerges from the lens of Bayesian optimal experimental design (BOED) (Foster et al., 2021; Rainforth et al., 2024), which prescribes selecting actions that maximize the *expected information gain* (EIG) over an unknown target variable (Panousis and Rainforth, 2022). While BOED has been widely applied in active learning (Melo et al., 2024) and Bayesian optimization (Liu et al., 2024), it remains under-explored in the context of LLM-based reasoning, where each token or block can be viewed as a design decision (Handa et al., 2024; Falck et al., 2024; Xiao et al., 2025; Qiu et al., 2025). In contrast, we propose to view each reasoning block as a sequential design variable whose utility can be approximated by its marginal reduction in uncertainty over the final answer. By adapting the BOED principle to CoT generation, our approach connects confidence estimation and trajectory control under a single theoretical framework. To the best of our knowledge, CaR is the first reasoning-time framework to operationalize BOED for stepwise generation, bridging the gap between information-theoretic decision making and LLM decoding.

7 Conclusion

We present CaR, a unified self-guided framework for optimizing the reasoning trajectories of LLMs at inference time. By approximating information gain with token-level confidence scores, CaR enables two complementary interventions: low-confidence block pruning and high-confidence early stopping. Our method operates entirely at the decoding level without requiring model modifications or auxiliary training, making it broadly compatible with future reasoning models. Future directions include extending CaR beyond mathematical reasoning and enhancing its robustness.

Limitations

While our method introduces a principled and efficient framework for reasoning trajectory control, it has several limitations. First, our confidence estimates are based solely on token-level log-probabilities from a single forward pass, which may not fully capture epistemic uncertainty—particularly in ambiguous or multi-answer settings. Second, the approach assumes that reasoning blocks are cleanly segmented via delimiters (e.g., “Wait”, “hmm”, “Alternatively”), an assumption that may not hold in domains or models lacking explicit CoT formatting. Third, although CaR demonstrates strong performance across a range of open-source models, it has not yet been evaluated on closed or instruction-tuned systems with different output conventions. Finally, the method does not explicitly address hallucinated or logically inconsistent intermediate steps, which remain a broader challenge for decoding-time reasoning control.

An important additional limitation of CaR lies in its reliance on the language model’s internal confidence, as approximated by token-level likelihoods. Large language models are known to exhibit overconfidence or miscalibrated uncertainty, especially on out-of-distribution queries or complex multi-hop reasoning. In the absence of external supervision or verification signals, confidently incorrect blocks may be retained or prematurely trigger final answer emission. While CaR partially mitigates this via threshold tuning and resampling, future work could incorporate stronger uncertainty estimation, calibration techniques, or auxiliary self-verification modules. Beyond model-internal refinements, another direction involves combining CaR with retrieval-augmented reasoning or consistency-based methods to further improve robustness on open-ended tasks.

References

- Simon A Aytes, Jinheon Baek, and Sung Ju Hwang. 2025. [Sketch-of-thought: Efficient llm reasoning with adaptive cognitive-inspired sketching](#). *arXiv preprint arXiv:2503.05179*.
- Kathryn Chaloner and Isabella Verdinelli. 1995. Bayesian experimental design: A review. *Statistical science*, pages 273–304.
- Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024. Do not think that much for $2+3=?$ on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.
- Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, and 1 others. 2025. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*.
- DeepSeek-AI. 2025. [Deepseek-r1: A high-performance open-source reasoning model](#). *arXiv preprint arXiv:2504.20708*.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint, arXiv:2501.12948*.
- Fabian Falck, Ziyu Wang, and Christopher C Holmes. 2024. Are large language models bayesian? a martingale perspective on in-context learning. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.
- Chenrui Fan, Ming Li, Lichao Sun, and Tianyi Zhou. 2025. Missing premise exacerbates overthinking: Are reasoning models losing critical thinking skill? *arXiv preprint arXiv:2504.06514*.
- Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. Efficient reasoning models: A survey. *arXiv preprint arXiv:2504.10903*.
- Adam Foster, Johannes Beck, and Tom Rainforth. 2021. [Variational bayesian optimal experimental design](#). *arXiv preprint arXiv:2102.05274*.
- Adam Foster, Martin Jankowiak, Elias Bingham, Paul Horsfall, Yee Whye Teh, Thomas Rainforth, and Noah Goodman. 2019. Variational bayesian optimal experimental design. *Advances in Neural Information Processing Systems*, 32.
- 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*.
- Hasan Abed Al Kader Hammoud, Hani Itani, and Bernard Ghanem. 2025. Beyond the last answer: Your reasoning trace uncovers more than you think. *arXiv preprint arXiv:2504.20708*.
- Kunal Handa, Yarin Gal, Ellie Pavlick, Noah Goodman, Jacob Andreas, Alex Tamkin, and Belinda Z Li. 2024. Bayesian preference elicitation with language models. *arXiv preprint arXiv:2403.05534*.

- 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*.
- 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*.
- Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. 2024. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. *arXiv preprint arXiv:2401.10480*.
- Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, and 1 others. 2025. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*.
- Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mhaela van der Schaar. 2024. Large language models to enhance bayesian optimization. *arXiv preprint arXiv:2402.03921*.
- Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025a. [O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning](#). *arXiv preprint arXiv:2501.12570*.
- Yijia Luo, Yulin Song, Xingyao Zhang, Jiaheng Liu, Weixun Wang, GengRu Chen, Wenbo Su, and Bo Zheng. 2025b. Deconstructing long chain-of-thought: A structured reasoning optimization framework for long cot distillation. *arXiv preprint arXiv:2503.16385*.
- Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025. [Nothinking: Re-thinking the role of explicit reasoning in llms](#). *arXiv preprint arXiv:2504.09858*.
- Rohin Manvi, Anikait Singh, and Stefano Ermon. 2024. Adaptive inference-time compute: LLMs can predict if they can do better, even mid-generation. *arXiv preprint arXiv:2410.02725*.
- Luckeciano C Melo, Panagiotis Tigas, Alessandro Abate, and Yarin Gal. 2024. Deep bayesian active learning for preference modeling in large language models. *arXiv preprint arXiv:2406.10023*.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.
- Tergel Munkhbat, Namgyu Ho, Seo Hyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. 2025. Self-training elicits concise reasoning in large language models. *arXiv preprint arXiv:2502.20122*.
- Emmanouil Panousis and Tom Rainforth. 2022. [A principled and unified framework for bayesian optimal experimental design](#). *arXiv preprint arXiv:2202.04647*.
- Linlu Qiu, Fei Sha, Kelsey Allen, Yoon Kim, Tal Linzen, and Sjoerd van Steenkiste. 2025. Bayesian teaching enables probabilistic reasoning in large language models. *arXiv preprint arXiv:2503.17523*.
- Tom Rainforth, Adam Foster, Desi R Ivanova, and Freddie Bickford Smith. 2024. Modern bayesian experimental design. *Statistical Science*, 39(1):100–114.
- 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*.
- Jinyan Su, Jennifer Healey, Preslav Nakov, and Claire Cardie. 2025. Between underthinking and overthinking: An empirical study of reasoning length and correctness in llms. *arXiv preprint arXiv:2505.00127*.
- Qwen Team. 2025. [Qwq-32b: Embracing the power of reinforcement learning](#).
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. [Chain-of-thought prompting elicits reasoning in large language models](#). *arXiv preprint arXiv:2201.11903*.
- Yuyang Wu, Yifei Wang, Tianqi Du, Stefanie Jegelka, and Yisen Wang. 2025. When more is less: Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*.
- Xiao Xiao, Yu Su, Sijing Zhang, Zhang Chen, Yadong Chen, and Tian Liu. 2025. Confidence in large language model evaluation: A bayesian approach to limited-sample challenges. *arXiv preprint arXiv:2504.21303*.
- Fengli Xu, Qianye Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, and 1 others. 2025a. [Towards large reasoning models: A survey of reinforced reasoning with large language models](#). *arXiv preprint arXiv:2501.09686*.
- Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025b. [Chain of draft: Thinking faster by writing less](#). *arXiv preprint arXiv:2502.18600*.
- Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Zheng Lin, Li Cao, and Weiping Wang. 2025. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*.

- Jingyang Yi and Jiazheng Wang. 2025. Shorterbetter: Guiding reasoning models to find optimal inference length for efficient reasoning. *arXiv preprint arXiv:2504.21370*.
- Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025. Reasoning models know when they’re right: Probing hidden states for self-verification. *arXiv preprint arXiv:2504.05419*.
- Zirui Zhao, Hanze Dong, Amrita Saha, Caiming Xiong, and Doyen Sahoo. 2024. Automatic curriculum expert iteration for reliable llm reasoning. *arXiv preprint arXiv:2410.07627*.

A Additional Discussion about BOED

We draw theoretical motivation from the framework of Bayesian Optimal experimental design (BOED) (Foster et al., 2019), which formalizes how to make optimal decisions under uncertainty by acquiring the most informative observations (Chaloner and Verdinelli, 1995). The central objective is to select a design variable ξ that maximizes the expected utility of an experiment, often instantiated as the **expected information gain (EIG)** about a latent target variable θ (Rainforth et al., 2024). Formally, EIG quantifies the reduction in uncertainty about θ upon observing data y under a given design ξ , and is defined as the mutual information:

$$\text{EIG}_\theta(\xi) = \mathbb{E}_{p(\theta, y|\xi)} \left[\log \frac{p(y|\theta, \xi)}{p(y|\xi)} \right], \quad (8)$$

which corresponds to the expected KL divergence between the posterior $p(\theta | y, \xi)$ and the prior $p(\theta)$.

BOED provides a general principle for **sequential information acquisition**, which motivates us to reinterpret each reasoning block as a design decision, and the final answer as the latent target variable. From this perspective, the goal of reasoning becomes selecting the next block to *maximize its expected information gain* to the final answer. Intuitively, each step should contribute meaningfully to reducing predictive uncertainty, and reasoning should proceed only as long as additional information is worth acquiring. This interpretation approximates the marginal utility of each block by reducing uncertainty over the answer distribution.

B Additional Experiment Details

B.1 Reasoning Models

We evaluate our confidence-aware reasoning framework on a diverse set of large reasoning language models (LRLMs), including the DeepSeek-R1-Distill-Qwen family (1.5B, 7B, 14B) and the QwQ-32B model. The DeepSeek series represents instruction-tuned models optimized for chain-of-thought (CoT) generation, with explicit reasoning-answer delimiters. QwQ-32B serves as a high-capacity baseline with strong reasoning performance across math and logic benchmarks. All models are evaluated in a zero-shot setting with consistent CoT-style prompts and no additional fine-tuning or calibration, allowing us to isolate the effect of our decoding-time intervention.

B.2 Datasets and benchmarks

We evaluate our method on four benchmark datasets that emphasize multi-step reasoning, symbolic manipulation, and factual knowledge retrieval. To assess mathematical problem-solving capabilities, we include AMC 2023 (American Mathematics Competitions) and AIME 2023–2025 (American Invitational Mathematics Examination), both consisting of high-school level problems with non-trivial algebraic and combinatorial structure. These datasets challenge models to perform precise symbolic reasoning over several steps, making them ideal for evaluating reasoning trajectory optimization. In addition, we include GPQA Diamond recently proposed benchmark for graduate-level professional question answering—which tests factual recall, deductive reasoning, and domain-specific knowledge across fields such as law, medicine, and engineering. To assess generalization under token budget constraints, we also evaluate on a filtered subset of MATH-500, a collection of moderately difficult open-domain math problems designed to test both CoT depth and correctness. Across all datasets, we follow prior work in adopting a zero-shot chain-of-thought prompting setup, and we report exact match accuracy on the final answers.

B.3 Performance Metrics

Our evaluation focuses on two complementary objectives: reasoning accuracy and computational efficiency. Following our formulation in Section 4, we aim to maximize the quality of the final answer while minimizing the length of the reasoning trajectory. Accordingly, we report the following metrics:

- **Answer Accuracy:** the primary evaluation metric is exact match (EM) between the predicted answer and the gold label, evaluated under the model’s emitted answer format (e.g., `\boxed{}` or final token span). For multiple-choice problems (e.g., GPQA), EM is computed over the selected choice token.
- **Reasoning Length:** to quantify inference-time efficiency, we measure the total number of tokens generated in the reasoning phase prior to answer emission. This includes all tokens between the start of the slow-thinking region and the point of early stopping (if triggered). Tokens from the final answer are excluded.

B.4 Implementation Details

We use greedy decoding with a maximum generation length of 16,384 tokens for all models. The generation temperature is set to 0.6 by default to encourage determinism while maintaining fluency. For low-confidence block pruning (Section 3.2), we resample reasoning blocks using a higher temperature of 0.7 to promote diversity, and retain the candidate with the highest confidence among $K = 3$ samples. Thresholds for confidence-based control are fixed across all models and datasets. We set the low-confidence pruning threshold to $\tau_{\text{low}} = 0.55$, such that blocks below this value are discarded and regenerated. For early stopping (Section 3.3), we apply a two-stage gating procedure: a block-level threshold $\tau_{\text{high}} = 0.8$ determines when to trigger candidate answer generation, and an answer-level threshold $\tau_{\text{ans}} = 0.95$ is used to decide whether to halt reasoning and emit the final output. These values were chosen based on validation performance on held-out samples from the AMC dataset and held fixed across all experiments.

C Additional Experiments and Analysis

C.1 More Baseline Comparisons

We have conducted additional experiments, including two additional baseline methods:

- Majority Voting ($k=5$): Generates $k=5$ reasoning chains and selects the most frequent final answer, following standard self-consistency setups.
- Token-Conditional Control (TCC) (Muennighoff et al., 2025): Specifies a fixed token budget in the system prompt to enforce concise reasoning. We set this token limit based on the average length generated by CaR to ensure a fair comparison.

The results are summarized below (AMC 2023, GPOQA-Diamond, AIME 2023) and the key observations are:

- CaR consistently outperforms both Majority Voting and TCC across all datasets and model scales.
- Compared to TCC, CaR achieves +2.5 to +3.3 accuracy gains, demonstrating that purely enforcing a token budget without confidence-aware reasoning is suboptimal.
- Compared to Majority Voting, CaR improves accuracy by +2.0 to +4.3, while also reducing average tokens due to pruning and early stopping.

C.2 Budget Compliance.

We further evaluate token budget compliance, defined as the proportion of examples whose reasoning phase terminates within a fixed maximum token length (e.g., 2048 tokens). While the CoT-Raw baseline frequently exceeds this threshold, particularly on arithmetic-heavy datasets such as AIME and GPQA. CaR significantly increases compliance rates across all settings. The results highlight CaR’s practical utility as a plug-and-play decoding strategy that can maintain high performance under bounded compute. For instance, on Distill-Qwen-7B, the proportion of inputs completed within budget increases from 62% to 94% on AIME-2025, and from 58% to 91% on GPQA-Diamond. This budget-awareness is particularly valuable for real-world applications where latency, API quota, or memory constraints limit the allowable generation cost per query.

C.3 Effect of Resampling during Pruning.

During pruning, CaR optionally resamples multiple candidate reasoning blocks when the initial one has low confidence. This introduces a trade-off between selection quality and decoding cost. Table 4 presents results on AMC-2023 (Distill-Qwen-7B) with varying resampling counts $K \in \{1, 3, 5\}$. Increasing K from 1 to 3 substantially improves accuracy from 78.5% to 85.0% ($\uparrow 6.5$) and reduces reasoning length by 9%, which is a strong gain relative to only 1.7 $\times$ decoding overhead. Further increasing K to 5 yields a marginal +0.5 accuracy gain at a significantly higher cost (2.5 $\times$), suggesting diminishing returns. We therefore fix $K = 3$ throughout all experiments as a practical compromise, achieving near-maximum performance while keeping compute overhead moderate.

C.4 Trajectory Length Stability.

To better understand how CaR adapts its reasoning depth across tasks and models, we measure the average number of reasoning blocks generated before emitting a final answer. As shown in Figure 7, CaR maintains a stable and balanced number of blocks across all six datasets and three model scales (1.5B, 14B, and 32B). For example, on challenging tasks like MATH-500 and GPQA-Diamond, the average number of blocks remains below 50 even for smaller models, while on shorter-answer tasks such as AIME-2023, CaR converges in fewer than 20 blocks. Importantly, CaR exhibits

Model	Method	AMC	GPQA	AIME 2023
Distill-Qwen-7B	CoT-Raw	75.0	24.8	40.0
Distill-Qwen-7B	TCC (Muennighoff et al., 2025)	82.5	27.3	46.7
Distill-Qwen-7B	Majority Voting (k=5)	78.0	25.7	43.3
Distill-Qwen-7B	CaR	85.0	28.3	50.0
Distill-Qwen-14B	CoT-Raw	82.5	52.0	73.3
Distill-Qwen-14B	TCC (Muennighoff et al., 2025)	88.5	58.3	76.7
Distill-Qwen-14B	Majority Voting (k=5)	84.5	55.8	73.3
Distill-Qwen-14B	CaR	90.0	59.1	80.0

Table 3: Performance comparison across different baselines and models.

# of Samples	Accuracy $\uparrow$	Token Length $\downarrow$	Relative Cost
K=1	78.5	1693	1 $\times$
K=3	85.0	1539	1.7 $\times$
K=5	85.5	1512	2.5 $\times$

Table 4: Effect of the number of resampling in low-confidence reasoning block pruning.

consistent behavior across models, suggesting that its confidence-based control mechanism scales robustly with model capacity. These results highlight CaR’s ability to dynamically adjust reasoning depth in a content-aware manner, without over-generating or prematurely stopping. The uniformity across tasks further demonstrates the framework’s generality and its suitability for deployment under varied reasoning demands.

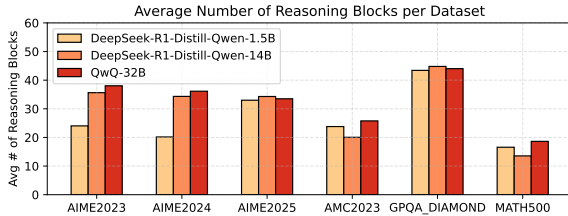


Figure 7: Converged number of reasoning blocks across different LRMs and benchmarks.

C.5 How does CaR impact the inference latency?

Overall effect on latency. Our experiments show that CaR reduces end-to-end inference latency, primarily because:

- Pruning removes low-confidence reasoning blocks early.
- Early stopping halts generation once confidence is sufficiently high.

Latency is measured as wall-clock time, including decoding and post-processing, on an A100-40GB GPU.

Computational overhead analysis. While CaR reduces generation length and latency, it introduces minimal additional computation:

Engineering considerations.

- CaR does not require additional forward passes or cross-process communication.
- All pruning and stopping decisions are implemented inline within the decoding loop, ensuring minimal orchestration delays.

Overall, CaR reduces inference latency by shortening generation length, with negligible computational overhead from confidence calculation or decision logic. Compared to SC, CaR offers a much more latency-efficient improvement pathway. We will clarify these latency trade-offs and engineering considerations in the camera-ready version to better inform deployment implications.

C.6 Segment Reasoning Blocks

In our current implementation, we target reasoning models such as DeepSeek and Qwen series models, which reliably generate explicit block markers (e.g., “Step i:”, “Wait:”, “Hmm”) when prompted appropriately. Empirically, >98% of generations include such markers, enabling straightforward segmentation via string matching. However, we acknowledge that if these explicit signals are absent, segmentation becomes non-trivial and is a limitation of the current CaR framework (as noted in §7 Limitations).

To address this, we are exploring automatic block segmentation methods, including:

- Perplexity-based segmentation: identify split points where token-level perplexity sharply changes.
- Sentence-level semantic clustering: embed token sequences and segment when adjacent embeddings show low similarity.

Component	Overhead impact
Confidence calculation	Negligible. Uses token logprobs already produced by the decoder; adds < 1 ms per block for aggregation.
Pruning & early stopping logic	Negligible. Involves simple threshold comparisons.
Block segmentation, parsing	Minimal. Uses string matching for explicit markers (e.g., “Step i:”); fallback heuristics (newline + punctuation) incur low processing cost.
Prompt size change	Minimal. Including pruning/stopping instructions slightly increases prompt tokens but does not meaningfully affect decode speed.

Table 5: Computational overhead introduced by CaR components.

- Lightweight learned segmentation classifiers: train a small classifier to predict block boundaries using annotated CoT samples.
- If no segmentation is possible, CaR can fallback to fixed-length token windows as an approximate solution, albeit with reduced interpretability.

C.7 Relationship Between Overthinking and Confidence

Our CaR’s central hypothesis is that overthinking correlates with declining token-level confidence, and we provide both empirical evidence and new analyses to support this.

Low-confidence and pruning. As shown in Figures 5 and 6, reasoning blocks with confidence < 0.6 often contain transitional or filler phrases such as “wait, no...”, which typically do not contribute meaningfully to the final reasoning outcome. We acknowledge that some low-confidence blocks may still be helpful, though their proportion is small.

To quantify this, we conducted a new annotation experiment on the AIME 2023–2025 datasets. We sampled reasoning blocks with confidence < 0.6 and used an O3 model to label each as **Helpful** (contributing to correct reasoning) or **Redundant/Harmful**. Then we conduct a manual inspection to verify the labels for accuracy.

These results in Table 6 indicate that only $\sim 5\%$ of low-confidence blocks are actually helpful, confirming that pruning low-confidence blocks rarely removes useful reasoning steps. Moreover, pruning improves both accuracy and efficiency, further suggesting that these blocks are less helpful on average.

High-confidence and early stopping. We agree that high-confidence blocks can still contain hallucinations, as noted in §7 Limitations. However, in practice, models perform well when early stopping is triggered by high-confidence thresholds, as also

observed in recent RL-based reasoning optimization studies.

To validate this, we conducted an additional block annotation experiment. Using the O3 model, we labeled reasoning blocks as **Helpful** or **Redundant**, and calculated their average confidence scores (plus the standard deviation). These results in Table 7 show that **Helpful blocks consistently have higher confidence** than redundant ones, supporting our use of high-confidence early stopping as a practical heuristic.

D Prompt Format Examples

We show a representative prompt used for inference with Distill-Qwen-7B on an AIME problem:

Question: Let x be a positive real number such that $x + \frac{1}{x} = 3$. What is the value of $x^2 + \frac{1}{x^2}$?
<think> Let’s first square both sides of the equation... </think>

Reasoning blocks are automatically segmented at the “Wait” delimiter during decoding.

E Hyperparameter Selection

We sweep the following values for threshold tuning (§6.1):

- $\tau_{\text{low}} \in \{0.50, 0.52, 0.54, 0.55, 0.56, 0.58, 0.60\}$
- $\tau_{\text{high}} \in \{0.7, 0.75, 0.80, 0.85, 0.90\}$
- $\tau_{\text{ans}} \in \{0.90, 0.95, 1.00\}$
- $K \in \{1, 3, 5\}$

We fix $\tau_{\text{low}} = 0.55$, $\tau_{\text{high}} = 0.80$, and $\tau_{\text{ans}} = 0.95$ in all final experiments.

In our current experiments, we performed coarse grid searches and found that:

- For Qwen-series models, any $\tau_{\text{low}} \in [0.55, 0.65]$ and $\tau_{\text{high}} \in [0.70, 0.90]$ yields accuracy variation within ± 0.5 .

Dataset	Model	% Helpful (confidence <0.6)
AIME 2023	1.5B	5.8%
AIME 2024	1.5B	4.2%
AIME 2025	1.5B	3.5%
AIME 2023	7B	4.9%
AIME 2024	7B	4.3%
AIME 2025	7B	4.7%

Table 6: Percentage of helpful blocks among low-confidence (<0.6) blocks.

Dataset	Helpful (avg confidence)	Redundant (avg confidence)
AIME 2023	0.82 (± 0.04)	0.58 (± 0.08)
AIME 2024	0.84 (± 0.06)	0.53 (± 0.10)
AIME 2025	0.85 (± 0.07)	0.55 (± 0.09)

Table 7: Average confidence scores for Helpful vs. Redundant blocks.

Symbol	Description
$\mathcal{P}$	Problem prompt or input question
$\mathcal{L}_\theta$	Language reasoning model with parameters θ
$\mathcal{R}_{<t}$	Partial reasoning trajectory before step t
r_t	Candidate reasoning block at step t
$\mathcal{R}$	Full reasoning trajectory up to final block r_t
$\mathcal{A}_t$	Candidate answer generated after block r_t
$\mathcal{A}$	Final predicted answer
$\mathcal{O} = (\mathcal{R}, \mathcal{A})$	Complete model output: trajectory and answer
$\text{Conf}(\cdot)$	Confidence score based on average token log-probability
$\tau_{\text{low}}, \tau_{\text{high}}, \tau_{\text{ans}}$	Confidence thresholds for pruning and early stopping
K	Number of samples used for block resampling

Table 8: Summary of notations used in CaR .

- Preliminary tests on **DeepSeek-R1-Distill-Llama-8B** confirm similar stability ranges, suggesting that these thresholds generalize well across architectures, see our ablation studies in Section 4.3. Moreover, tuning these thresholds is computationally lightweight, requiring only evaluation on a small held-out dev set without additional backpropagation.

In future work, we plan to explore adaptive thresholding strategies, such as percentile-based calibration, selecting τ based on confidence distribution quantiles.

F Notation Summary

For clarity and reproducibility, we summarize the key notations used throughout the paper. Our framework operates at the decoding level of large reasoning models (LRMs), where each reasoning step is treated as an action in a self-guided trajectory.