# DR.COT: DIAGNOSING AND REGULATING CHAIN-OF-THOUGHT VIA SPARSE FEATURE COVERAGE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Large reasoning models improve complex problem solving through extended chain-of-thought (CoT) reasoning, but excessive reasoning increases inference cost and can even degrade accuracy, a phenomenon known as overthinking. Existing approaches mitigate overthinking by estimating reasoning sufficiency from trajectory-derived signals, yet lack explicit prompt-conditioned modeling of the reasoning state, limiting their ability to diagnose sufficiency and realign the trajectory when reasoning deviates from the prompt. To address these limitations, we propose **Dr.CoT**, a framework for diagnosing and regulating chain-of-thought reasoning via prompt-conditioned sparse feature coverage. Specifically, Dr.CoT first represents the prompt and reasoning trajectory in a shared sparse feature space, and uses prompt-conditioned feature coverage as an indicator of the current reasoning state. It then tracks feature coverage and its gain to diagnose reasoning sufficiency, adaptively stopping reasoning when coverage is sufficient and gains remain persistently low, which improves the accuracy–efficiency trade-off. During continued reasoning, it further uses uncovered prompt features to steer the model toward addressing the prompt, correcting deviations of the reasoning trajectory from the prompt. Extensive experiments across three model scales show that Dr.CoT reduces average token usage by **17.01%** while improving average accuracy by **2.16%** over standard CoT, and also outperforms all evaluated baselines.

## 1 INTRODUCTION

Large reasoning models (LRMs) improve complex problem solving through extended chain-of-thought (CoT) reasoning (Wei et al., 2022). Longer CoTs allow LRMs to leverage additional test-time compute, improving performance on challenging reasoning tasks in domains such as mathematics and science (Guo et al., 2025; Muennighoff et al., 2025). However, LRMs often struggle to determine when reasoning is sufficient and therefore continue reasoning (Wu et al., 2025). This can introduce repetition, unnecessary verification, and reasoning drift (Huang et al., 2025; Caldarella et al., 2026), increasing inference cost and potentially degrading accuracy (Yang et al., 2025; Kim et al., 2026). This phenomenon is commonly referred to as *overthinking*. Mitigating overthinking is therefore essential for improving inference efficiency without sacrificing reasoning accuracy.

Existing approaches mainly mitigate overthinking by estimating reasoning sufficiency from trajectory-derived signals. These signals capture different aspects of the reasoning trajectory, including confidence-based signals (Yang et al., 2026; Fu et al., 2026), reflective and semantic redundancy (Sun et al., 2026; Min et al., 2026), entropy dynamics (Xu et al., 2026), and model-generated sufficiency assessments (Xiang et al., 2026). Rather than relying on a fixed thinking budget, these methods adapt the stopping point according to the observed reasoning trajectory. The estimated sufficiency of the current reasoning is then used to determine whether reasoning should continue or stop.

Despite their effectiveness, existing methods remain limited in how reasoning progress is represented, assessed, and regulated. **(1) Limited prompt-conditioned modeling of reasoning progress.** Most existing methods assess reasoning progress from trajectory-derived signals, without an explicit prompt-conditioned measure of progress (Li et al., 2026; Xu et al., 2026; Wu et al., 2025; Sun et al., 2026). Even approaches that consider both the prompt and generated CoT (Xiang et al., 2026) do not explicitly distinguish which prompt-relevant information has been addressed and which remains unresolved. **(2) Limited explicit diagnosis of reasoning sufficiency.** Existing stopping criteria



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.

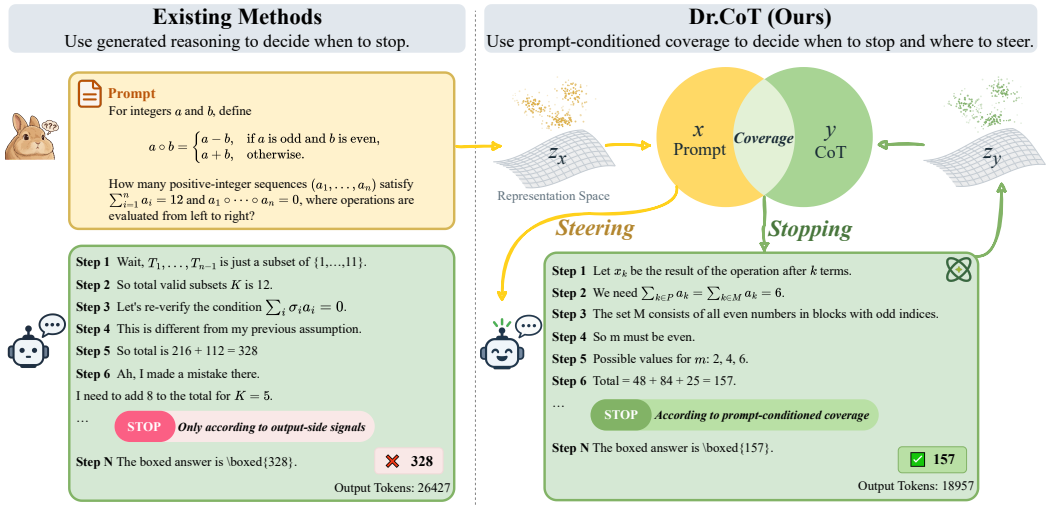


Figure 1: **Comparison between existing methods and Dr.CoT.** Existing methods estimate reasoning sufficiency from trajectory-derived signals to determine when to stop. In contrast, Dr.CoT introduces prompt-conditioned feature coverage, using coverage dynamics for *adaptive stopping* and uncovered prompt features for *targeted steering*.

infer reasoning sufficiency from trajectory-derived signals, yet these signals do not always reliably indicate whether sufficient reasoning has been reached. Models may become prematurely confident or appear stable before reasoning is complete, while temporary stagnation can be mistaken for true completion (Gai et al., 2026; Min et al., 2026). Therefore, determining whether the current reasoning is genuinely sufficient remains challenging. **(3) Limited active correction of reasoning drift.** Early-stopping methods mainly determine whether reasoning should continue or stop, but have limited ability to correct the trajectory when reasoning deviates from the prompt. Although activation steering can modify internal representations (Turner et al., 2023; He et al., 2026; Fang et al., 2026), its steering directions are typically not derived from the unresolved information of the current problem (Chen et al., 2025; Li et al., 2026; Huang et al., 2025).

To address these challenges, we propose **Dr.CoT**, a framework for diagnosing and regulating chain-of-thought reasoning through sparse feature coverage. As illustrated in Figure 1, Dr.CoT introduces prompt-conditioned feature coverage to support both adaptive stopping and targeted steering. Specifically, Dr.CoT first represents the prompt and reasoning trajectory in a shared sparse feature space using a sparse autoencoder (SAE) (Huben et al., 2024; Templeton et al., 2024), where prompt features define a problem-specific reference, while accumulated CoT features summarize the reasoning state reached so far. It then measures the coverage of prompt features by the accumulated CoT and tracks its gain to diagnose reasoning sufficiency. Reasoning stops when coverage is sufficient and its gain remains persistently low, helping avoid premature stopping caused by temporary stagnation. During continued reasoning, uncovered prompt features are decoded into a residual-stream direction to steer the model toward unresolved prompt-relevant information and correct reasoning drift. Together, these mechanisms enable Dr.CoT to improve the accuracy–efficiency trade-off and realign reasoning trajectories with the prompt, without task-specific fine-tuning.

Extensive experiments across four benchmarks and three model scales show that Dr.CoT reduces average token usage by **17.01%** while improving average accuracy by **2.16%** over standard CoT, and consistently outperforms the evaluated baselines. In summary, our main contributions are:

- Prompt-conditioned sparse feature coverage is introduced to explicitly quantify reasoning progress by measuring how the generated CoT covers prompt-relevant information.
- Coverage-guided sufficiency diagnosis is proposed based on feature coverage and sustained coverage gain, enabling adaptive stopping while mitigating premature stopping.
- Residual feature steering is developed to use uncovered prompt features to guide reasoning, helping avoid deviations of the reasoning trajectory from the prompt.

## 2 Related work

**Mitigating Overthinking in Large Reasoning Models** Existing overthinking mitigation methods broadly fall into training-based and inference-time paradigms. Training-based methods mitigate overthinking through adaptive reasoning modes (Zhang et al., 2025; Fang et al., 2025), budget-aware reasoning optimization (Luo et al., 2026; Shen et al., 2025; Wen et al., 2025), and CoT compression (Xia et al., 2025; Ma et al., 2025). Process-level approaches further reduce redundant reasoning via truncated-rollout reward shaping (Dai et al., 2025), semantic-step optimization (Chen et al., 2026), decoupled credit assignment (Jiang et al., 2026), and sufficiency-guided reasoning (Wang et al., 2026). Inference-time methods reduce redundant reasoning through concise prompting and reflection-token suppression (Xu et al., 2025; Wang et al., 2025), or confidence-based early exit and trajectory filtering (Yang et al., 2026; Fu et al., 2026). Others determine when to stop based on reasoning sufficiency, reflective redundancy, or semantic convergence (Xiang et al., 2026; Sun et al., 2026; Min et al., 2026). Despite these advances, existing methods lack an explicit measure of how much prompt-relevant information has been addressed during reasoning. This motivates Dr.CoT to quantify reasoning progress through prompt-conditioned feature coverage.

**Activation Steering** Activation steering modifies internal representations at inference time to control model behavior. Early approaches derive steering directions from contrastive activation differences (Turner et al., 2023), while sparse autoencoders (SAEs) decompose hidden states into sparse features, enabling more fine-grained interventions (Huben et al., 2024; Templeton et al., 2024; Deng et al., 2026). Building on this, SAE-based methods support causal circuit editing (Marks et al., 2025) and targeted control of instruction following and reasoning behaviors (He et al., 2026; Fang et al., 2026). Activation steering has also been explored for reasoning efficiency using offline steering vectors or difficulty-adaptive modulation (Chen et al., 2025; Lin et al., 2025). More recently, ReBalance (Li et al., 2026) mitigates overthinking and underthinking by scaling or reversing a fixed steering vector according to model confidence. In contrast, Dr.CoT derives a problem-specific steering direction from uncovered prompt features to guide reasoning and reduce deviations from the prompt.

## 3 Method

This section describes Dr.CoT, a framework for diagnosing and regulating chain-of-thought reasoning via prompt-conditioned sparse feature coverage. As shown in Figure 2, Dr.CoT represents the prompt and reasoning trajectory in a shared sparse feature space and quantifies reasoning progress through *Prompt-Conditioned Feature Coverage* (Section 3.1). *Coverage-Guided Sufficiency Diagnosis* combines the coverage level with sustained low coverage gain to determine when to stop reasoning (Section 3.2). We further introduce *Residual Feature Steering*, which uses uncovered prompt features to guide the model toward addressing the prompt (Section 3.3).

### 3.1 Prompt-Conditioned Feature Modeling

The purpose of reasoning is to resolve the complex problem specified by the prompt. Reasoning progress should therefore be assessed by how the evolving trajectory addresses prompt-relevant information. However, additional steps may repeat earlier reasoning while leaving some of this information unaddressed. We thus seek a shared feature representation that quantifies prompt-conditioned coverage and identifies uncovered prompt features.

To characterize prompt-conditioned feature coverage, we decompose the prompt features $F_x$ into covered and uncovered components with respect to the accumulated reasoning features $G_t$:

$$F_x = \underbrace{\min(F_x, G_t)}_{\text{covered component}} + \underbrace{[F_x - G_t]_+}_{\text{uncovered component}}, \tag{1}$$

where $\min(F_x, G_t)$ gives the feature-wise overlap between prompt and reasoning activations, and $[F_x - G_t]_+$ retains the positive differences corresponding to uncovered prompt features.

This formulation reflects the asymmetric relationship between the prompt and the reasoning trajectory. The prompt provides a fixed coverage reference, while the reasoning trajectory may introduce additional features needed to solve the problem. Accordingly, we seek to quantify how much of the prompt reference is covered by the accumulated reasoning and identify uncovered prompt features.

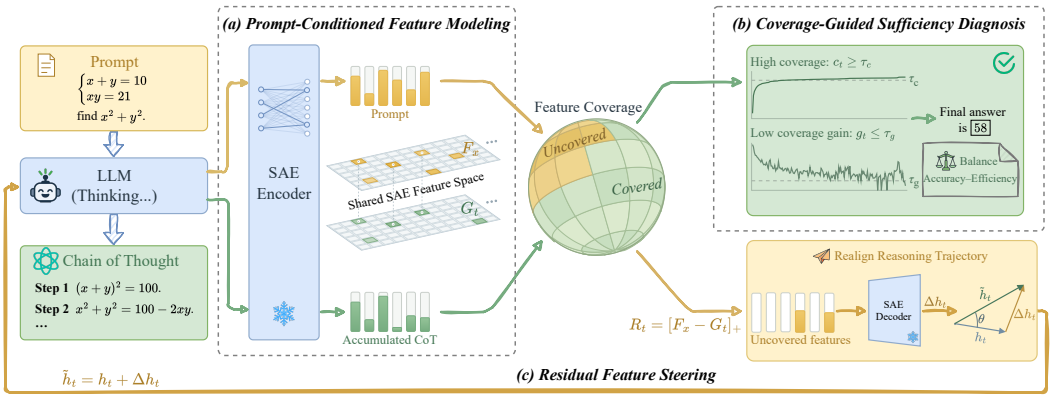


Figure 2: **Overview of Dr.CoT.** (a) *Prompt-Conditioned Feature Modeling* represents the prompt and reasoning trajectory in a shared sparse feature space to quantify reasoning progress through prompt-conditioned feature Modeling. (b) *Coverage-Guided Sufficiency Diagnosis* adaptively stops reasoning when coverage is sufficient and coverage gains remain persistently low. (c) *Residual Feature Steering* uses uncovered prompt features to construct a steering direction in the residual stream, guiding the model toward addressing the prompt.

Overall coverage and uncovered prompt features provide complementary information about the reasoning state. First, overall coverage measures how much of the prompt reference has been covered, but does not specify which features remain uncovered. Second, the uncovered component quantifies the remaining activation of each prompt feature. This motivates representing the prompt and reasoning trajectory in a shared sparse feature space.

To this end, we use an SAE to construct the shared sparse feature representation. Let $x = (x_1, \ldots, x_n)$ denote the tokenized input and $y_{1:t} = (y_1, \ldots, y_t)$ the generated reasoning prefix. At a selected transformer layer $\ell$, the frozen SAE encodes a residual-stream activation $h \in \mathbb{R}^d$ as

$$z(h) = \left[\mathrm{TopK}_K\left(W_{\mathrm{enc}} h + b_{\mathrm{enc}}\right)\right]_+, \tag{2}$$

where $d$ is the hidden dimension, $M$ is the number of SAE features, and $W_{\mathrm{enc}}$ and $b_{\mathrm{enc}}$ are the encoder weights and bias. $\mathrm{TopK}_K$ retains only the $K$ largest preactivations, and $[\cdot]_+$ retains only positive activations, leaving at most $K$ active features per token. The shared encoder enables feature-wise comparison between prompt and reasoning activations.

We define prompt and accumulated reasoning features as sums of token-level sparse activations:

$$F_x = \sum_{i=1}^{n} z(h_i^x), \qquad G_t = \sum_{s=1}^{t} z(h_s), \tag{3}$$

where $h_i^x$ and $h_s$ denote the residual-stream activations at layer $\ell$ for the $i$-th prompt token and the $s$-th reasoning token, respectively. The prompt features $F_x$ provide a fixed reference, while $G_t$ accumulates sparse feature activations over the first $t$ reasoning tokens. Their feature-wise comparison yields the covered and uncovered components in Equation 1, providing a shared basis for coverage-guided sufficiency diagnosis and residual feature steering.

### 3.2 Coverage-Guided Sufficiency Diagnosis

Dr.CoT diagnoses reasoning sufficiency from prompt feature coverage and its recent gain. Given the prompt feature state $F_x$ and accumulated reasoning state $G_t$, the uncovered feature residual is defined as $R_t = [F_x - G_t]_+$. For $\|F_x\|_1 > 0$, the corresponding coverage ratio is defined as

$$c_t = \frac{\sum_{j=1}^{M} \min(F_{x,j}, G_{t,j})}{\|F_x\|_1} = 1 - \frac{\|R_t\|_1}{\|F_x\|_1}. \tag{4}$$

Here, $R_t$ captures the remaining uncovered prompt-feature activation, while $c_t$ measures the fraction of prompt feature activation covered by the reasoning trajectory. The minimum operation ensures that

each feature contributes no more than its activation in the prompt, so repeatedly activating an already covered feature cannot compensate for prompt features that remain uncovered. Because the covered activation is capped by $F_x$ and normalized by $\|F_x\|_1$, $c_t$ is bounded in $[0, 1]$. Moreover, since SAE activations are nonnegative, $G_t$ can only increase as reasoning proceeds, making $c_t$ nondecreasing over time. Thus, $c_t$ serves as a prompt-conditioned measure of cumulative reasoning progress.

By itself, $c_t$ summarizes the cumulative coverage achieved so far, but does not indicate whether continued reasoning is still making coverage progress. Two reasoning trajectories can have similar coverage ratios but different dynamics: one may continue to cover unresolved prompt features, whereas the other may have nearly stopped gaining coverage. To distinguish these cases, we introduce the coverage gain $g_t$, which quantifies the increase in $c_t$. Formally, the coverage gain is defined as

$$g_t = \frac{c_t - c_{t-W}}{W}, \tag{5}$$

where $W$ denotes the window size in reasoning tokens. The gain $g_t$ measures the average increase in coverage per token over the recent $W$ reasoning tokens. Together, $c_t$ describes how much coverage has been accumulated, and $g_t$ captures the rate at which this coverage continues to grow. Low values of $g_t$ therefore indicate that the reasoning trajectory is making little coverage progress.

The coverage ratio $c_t$ and coverage gain $g_t$ play complementary roles in diagnosing reasoning sufficiency. A high $c_t$ indicates that substantial prompt-feature coverage has been achieved, while $g_t$ indicates whether continued reasoning is still making coverage progress. Accordingly, Dr.CoT diagnoses reasoning sufficiency using a joint criterion that combines sufficiently high coverage with persistently low coverage gain:

$$\begin{cases} c_t \geq \tau_c, \\ g_{t-kW} < \tau_g, \quad k = 0, \ldots, P-1. \end{cases} \tag{6}$$

where $\tau_c$ and $\tau_g$ denote the coverage and gain thresholds, respectively, and $P$ denotes the number of consecutive low-gain windows used to confirm persistent low coverage gain. By jointly considering $c_t$ and $g_t$, the criterion requires high prompt-feature coverage together with low coverage gain across multiple consecutive windows. Once this condition is met, Dr.CoT concludes the reasoning phase and proceeds to final answer generation, reducing unnecessary reasoning beyond sufficient coverage.

### 3.3 Residual Feature Steering

In addition to diagnosing reasoning sufficiency, Dr.CoT uses the uncovered feature residual $R_t$ to guide the ongoing reasoning trajectory toward the remaining uncovered prompt features. As illustrated in Figure 2(c), Dr.CoT decodes the uncovered feature residual into a steering direction through the SAE decoder. At each feature update step $u$, the direction is defined as

$$v_u = W_{\text{dec}} R_u + b_{\text{dec}}, \tag{7}$$

where $W_{\text{dec}}$ and $b_{\text{dec}}$ denote the SAE decoder weights and bias, respectively. Within each update interval, the same steering direction is applied to subsequent reasoning tokens. At the next feature update, newly accumulated reasoning features update $R_u$, and the steering direction is accordingly updated to reflect the prompt features that remain uncovered.

Given the steering direction $v_u$, Dr.CoT determines the intervention strength applied to each reasoning token. For a reasoning token $t$ in update interval $u$, $\Delta h_t$ denotes the additive perturbation to the residual-stream activation $h_t$. When $v_u \neq 0$, Dr.CoT constrains the perturbation as follows:

$$\Delta h_t^\star = \arg\max_{\Delta h} \langle \Delta h, v_u \rangle \quad \text{s.t.} \quad \|\Delta h\|_2 \leq \eta_t \|h_t\|_2. \tag{8}$$

where $\eta_t \in [0, \lambda]$ denotes the relative steering strength at token $t$. The perturbation is aligned with $v_u$, while its magnitude is scaled relative to $h_t$.

Accordingly, the perturbation is RMS-normalized to preserve the steering direction $v_u$ while adapting its magnitude to the current activation $h_t$. This normalization decouples the steering direction from the token-wise activation scale, giving the following form:

$$\Delta h_t^\star = \eta_t \frac{\text{RMS}(h_t)}{\text{RMS}(v_u)} v_u. \tag{9}$$

where $\mathrm{RMS}(\mathbf{a}) = \sqrt{\frac{1}{d}\sum_{j=1}^{d} a_j^2}$ denotes the root-mean-square magnitude over the hidden dimension.

The normalized perturbation is applied to the residual-stream activation as follows:

$$\widetilde{h}_t = h_t + \Delta h_t^{\star}. \tag{10}$$

The steered activation $\widetilde{h}_t$ then influences subsequent token generation, and the resulting reasoning features are accumulated into $G_t$. At the next feature update, the accumulated reasoning features $G_t$ are used to update the uncovered feature residual $R_{u+1}$ and the corresponding steering direction $v_{u+1}$. This closes the feedback loop between reasoning and steering: uncovered prompt features guide subsequent reasoning, while the resulting reasoning features update the uncovered feature residual. Through this feedback, Dr.CoT dynamically adapts the steering direction within a single reasoning trajectory, providing prompt-conditioned reasoning control without additional training.

## 4 Experiments

In this section, we present a comprehensive evaluation of Dr.CoT in terms of effectiveness, mechanisms, and practicality. We first evaluate its accuracy–efficiency trade-off across three model scales (Section 4.1), and then examine the mechanisms underlying coverage-guided sufficiency diagnosis and residual feature steering (Section 4.2). We further assess its practicality by analyzing the SAE training data budget and inference-time cost (Section 4.3). Additional analyses of sensitivity to sufficiency thresholds and steering strength are provided in Appendix E.

**Models and benchmarks.** We evaluate Dr.CoT with Qwen3.5-2B, Qwen3.5-9B, and Qwen3.5-27B (Qwen Team, 2026) using Qwen-Scope SAEs (Deng et al., 2026) on four benchmarks covering mathematical and scientific reasoning. For mathematical reasoning, we use AIME 2026 (30 problems) (Zhang & Math-AI, 2026) and HMMT February 2026 (33 problems) (Dekoninck et al., 2026). For scientific reasoning, we use GPQA-Diamond (198 questions) (Rein et al., 2023) and HLE-STEM (87 questions). In particular, HLE-STEM consists of text-only STEM multiple-choice questions selected from the Gold subset of HLE-Verified (Center for AI Safety et al., 2026; Zhai et al., 2026), enabling efficient and reproducible evaluation of STEM reasoning. Implementation and evaluation settings are detailed in Appendix C.1, with additional benchmark details provided in Appendix C.2.

**Baselines.** We compare Dr.CoT with Vanilla, NoThinking, and three representative methods for adaptive early stopping: DEER (Yang et al., 2026), REFRAIN (Sun et al., 2026), and CUSUM (Xu et al., 2026). Vanilla uses standard CoT generation, while NoThinking uses the model's native non-thinking mode. DEER identifies reasoning transitions, induces a trial answer, and exits when the trial answer reaches sufficiently high confidence. REFRAIN uses a two-stage stop discriminator to identify reflective redundancy and dynamically adjusts the stopping threshold according to reasoning characteristics. CUSUM detects the transition from a high-entropy uncertainty region to a low-entropy confidence region and triggers early exit once sufficient evidence of convergence is accumulated. All methods use the same answer extraction rules and evaluation criteria.

**Evaluation metrics.** We report accuracy (Acc, %) and generated tokens (Tokens) averaged over four samples per question. For efficiency evaluation, token usage is measured by the total number of generated tokens during inference, including reasoning, final answers, and any additional tokens introduced by the method. To summarize performance across tasks, Overall accuracy is averaged equally across the four benchmarks. Across backbones, we further report the average Overall accuracy and average token reduction compared with Vanilla. Finally, accuracy changes are reported in percentage points, and all reported averages are calculated before rounding.

### 4.1 Main Results

Dr.CoT consistently improves the accuracy–efficiency trade-off across model scales. As shown in Table 1, it reduces token consumption across models and benchmarks while improving overall accuracy on all three backbones. Overall, Dr.CoT reduces token consumption by **17.01%** and improves overall accuracy by **2.16%** (from 51.69% to 53.85%). We further analyze performance variations across model scales and comparisons with existing methods below.

Table 1: Main results across Qwen3.5 backbones: four-sample mean accuracy (Acc, %) and generated tokens. Overall accuracy is averaged equally across benchmarks; parentheses denote changes relative to Vanilla. Bold indicates the best accuracy per backbone; green shading indicates Dr.CoT.

| | AIME 2026 | | HMMT 2026 | | GPQA-Diamond | | HLE-STEM | | Overall | |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | Acc | Tokens | Acc | Tokens | Acc | Tokens | Acc | Tokens | Acc | Tokens |
| **Qwen3.5-2B** | | | | | | | | | | |
| Vanilla | 19.17 | 32,643 | 10.61 | 32,644 | **58.59** | 32,104 | 8.05 | 32,149 | 24.11 | 32,385 |
| NoThinking | 17.50 | 7,396 | 14.39 | 7,474 | 48.99 | 4,494 | 9.20 | 4,073 | 22.52 (-1.59) | 5,859 |
| DEER | 19.17 | 32,563 | 10.61 | 32,406 | 55.56 | 31,850 | 9.20 | 25,816 | 23.64 (-0.47) | 30,659 |
| REFRAIN | 18.33 | 31,738 | 11.36 | 31,694 | 54.55 | 26,088 | 13.79 | 25,431 | 24.51 (+0.40) | 28,738 |
| CUSUM | 15.00 | 23,159 | 11.36 | 21,844 | 50.51 | 8,293 | 12.64 | 7,389 | 22.38 (-1.73) | 15,171 |
| Dr.CoT | **20.83** | 22,887 | **17.42** | 25,470 | 57.58 | 20,978 | **14.94** | 19,414 | **27.69** (+3.59) | 22,187 |
| **Qwen3.5-9B** | | | | | | | | | | |
| Vanilla | 73.33 | 30,574 | **59.85** | 31,383 | 80.81 | 25,392 | **24.71** | 24,021 | 59.68 | 27,843 |
| NoThinking | 60.00 | 5,814 | 41.67 | 6,724 | 84.34 | 3,765 | 18.39 | 3,273 | 51.10 (-8.58) | 4,894 |
| DEER | 74.17 | 23,904 | 55.30 | 24,609 | 84.09 | 19,104 | 21.84 | 18,144 | 58.85 (-0.83) | 21,440 |
| REFRAIN | 76.67 | 26,994 | 53.79 | 27,352 | 80.81 | 16,911 | 23.85 | 15,362 | 58.78 (-0.90) | 21,655 |
| CUSUM | 69.17 | 13,150 | 52.27 | 15,233 | 81.82 | 7,250 | 22.70 | 7,625 | 56.49 (-3.19) | 10,815 |
| Dr.CoT | **81.67** | 27,247 | 56.82 | 27,664 | **85.35** | 23,083 | **24.71** | 20,855 | **62.14** (+2.46) | 24,712 |
| **Qwen3.5-27B** | | | | | | | | | | |
| Vanilla | **91.67** | 22,835 | 72.73 | 25,501 | 87.37 | 18,160 | 33.33 | 17,024 | 71.28 | 20,880 |
| NoThinking | 76.67 | 6,249 | 59.09 | 6,831 | 82.83 | 3,651 | 26.44 | 3,232 | 61.26 (-10.02) | 4,991 |
| DEER | **91.67** | 19,892 | 68.94 | 21,916 | **87.88** | 14,984 | 28.74 | 13,466 | 69.31 (-1.97) | 17,565 |
| REFRAIN | 90.83 | 20,867 | 70.45 | 23,061 | 86.36 | 13,247 | **36.78** | 11,706 | 71.11 (-0.17) | 17,220 |
| CUSUM | **91.67** | 13,333 | 70.45 | 19,426 | 83.84 | 13,908 | 34.48 | 14,341 | 70.11 (-1.17) | 15,252 |
| Dr.CoT | 90.83 | 20,603 | **74.24** | 23,121 | 87.37 | 16,407 | 34.48 | 16,454 | **71.73** (+0.46) | 19,146 |

The benefits of Dr.CoT vary across model scales, with larger accuracy gains observed on smaller backbones while token savings remain consistent. On Qwen3.5-2B, Dr.CoT achieves the largest improvement, reducing token consumption by 31.5% and improving overall accuracy by 3.59% compared with Vanilla. The improvement remains consistent on Qwen3.5-9B, where Dr.CoT reduces token consumption by 11.2% and improves overall accuracy by 2.46%. These gains extend across mathematical and scientific reasoning tasks: AIME accuracy increases from 73.33% to 81.67%, and GPQA-Diamond accuracy improves from 80.81% to 85.35%, both with shorter reasoning trajectories. On Qwen3.5-27B, although the accuracy gain becomes smaller due to the stronger Vanilla baseline, Dr.CoT still reduces token consumption by 8.3% while improving overall accuracy by 0.46%. Overall, these results show that Dr.CoT provides larger accuracy improvements on smaller backbones while consistently reducing reasoning cost across model scales.

Compared with existing early-stopping methods, Dr.CoT provides a better accuracy–efficiency trade-off across model scales. On Qwen3.5-2B, Dr.CoT uses fewer tokens than DEER and REFRAIN while achieving higher accuracy. On Qwen3.5-9B and Qwen3.5-27B, Dr.CoT is the only early-stopping method that improves accuracy over Vanilla. NoThinking and CUSUM achieve shorter generations at the cost of lower accuracy, suggesting that aggressive truncation may remove useful reasoning.

### 4.2 Mechanistic Analysis

**Component Contributions.** We independently ablate coverage-guided sufficiency diagnosis and residual feature steering on Qwen3.5-9B to analyze the contribution of each component, with results summarized in Table 2. Without coverage-guided sufficiency diagnosis, Dr.CoT generates longer reasoning trajectories. Compared with this variant, full Dr.CoT reduces token consumption by 9.8% while achieving comparable overall accuracy. Without residual feature steering, Dr.CoT retains sufficiency diagnosis. Compared with Vanilla, this variant reduces token consumption by 10.2% while achieving unchanged overall accuracy (59.68%). These results show that sufficiency diagnosis improves reasoning efficiency, while residual feature steering provides additional accuracy gains.

Table 2: Component ablations of Dr.CoT on Qwen3.5-9B. The w/o steering removes residual feature steering, while the w/o stopping removes coverage-guided sufficiency diagnosis. Acc (%) and Tokens are averaged over four samples per question; overall accuracy is averaged equally across benchmarks. Bold indicates the best accuracy; green highlights Dr.CoT.

| | AIME 2026 | | HMMT 2026 | | GPQA-Diamond | | HLE-STEM | | Overall | |
|---|---|---|---|---|---|---|---|---|---|---|
| Method | Acc | Tokens | Acc | Tokens | Acc | Tokens | Acc | Tokens | Acc | Tokens |
| w/o steering | 77.50 | 26,982 | 56.06 | 28,299 | 83.33 | 23,168 | 21.84 | 21,566 | 59.68 | 25,004 |
| w/o stopping | 80.00 | 30,154 | **60.61** | 30,898 | 81.82 | 25,338 | **25.86** | 23,143 | 62.07 | 27,383 |
| Full | **81.67** | 27,247 | 56.82 | 27,664 | **85.35** | 23,083 | 24.71 | 20,855 | **62.14** | 24,712 |

**Coverage-Guided Sufficiency Diagnosis.** On Qwen3.5-9B and AIME 2026, we evaluate the role of prompt-conditioned feature coverage by replacing the prompt feature representation $F_x$ with features extracted from another problem in the same benchmark, while keeping the sufficiency criterion unchanged. Steering is disabled in both settings to focus the analysis on the prompt feature reference used for evaluating accumulated reasoning features $G_t$. This experiment examines whether prompt-conditioned feature coverage is essential for reliable sufficiency diagnosis.

As shown in Table 3, the matched prompt reference achieves higher accuracy and shorter reasoning than the shuffled reference, improving accuracy by 5.00% and reducing token consumption by 10.9%. These results demonstrate the importance of prompt-conditioned feature coverage for reliable sufficiency diagnosis while keeping the input problem unchanged. The shuffled reference produces longer reasoning trajectories without recovering the matched reference performance, indicating that prompt-conditioned coverage is critical for identifying unresolved prompt features rather than simply extending reasoning for sufficiency diagnosis.

Table 3: Prompt reference ablation for sufficiency diagnosis on AIME 2026.

| Prompt reference | Acc (%) | Tokens |
|---|---|---|
| Matched prompt | 77.50 | 26,982 |
| Shuffled prompt | 72.50 | 30,268 |

**Residual Feature Steering.** Dr.CoT decodes the uncovered feature residual $R_u$ into the residual-stream direction $v_u$ (Equation 7). To evaluate the specificity of this direction, we compare it with reverse, random, full-prompt, and covered-feature steering. Reverse steering negates $v_u$, full-prompt steering decodes all prompt features, and covered-feature steering decodes only the features already addressed by the reasoning trajectory. These comparisons distinguish steering toward unresolved prompt-relevant information from reinforcing irrelevant or already covered features. All variants share the same injection location, schedule, and steering strength.

As shown in Table 4, residual feature steering achieves the highest accuracy of 81.67%, outperforming full-prompt steering (78.33%) and covered-feature steering (76.67%). Full-prompt steering uses slightly fewer tokens (0.6% reduction) but achieves lower accuracy, while random and reverse steering both degrade accuracy compared with no steering. These results indicate that steering toward unresolved prompt-relevant features is more effective than reinforcing the entire prompt or already covered features, suggesting that effective steering should focus on information not yet addressed by the reasoning trajectory. Compared with the variant without steering, residual feature steering improves AIME accuracy by 4.17%. Together, these results demonstrate the role of uncovered prompt features in effective residual feature steering.

Table 4: Steering direction ablation for residual feature steering on AIME 2026.

| Steering direction | Acc (%) | Tokens |
|---|---|---|
| w/o steering | 77.50 | 26,982 |
| Reverse | 74.17 | 27,843 |
| Random | 75.00 | 27,552 |
| Full prompt | 78.33 | 27,096 |
| Covered features | 76.67 | 28,155 |
| Uncovered features | 81.67 | 27,247 |

### 4.3 Practicality of Dr.CoT

To assess the practicality of Dr.CoT, we examine both the offline requirements and online costs introduced by the SAE component. Specifically, we analyze the sensitivity to SAE training data scale and evaluate the inference-time computational cost of Dr.CoT.

**SAE Training Budget.** To study the effect of SAE training budget, we vary the SAE training data from 0.1M to 10M tokens on Qwen3.5-9B. The SAEs are trained on a mixed corpus of verified mathematical reasoning traces from OpenR1-Math-220k (Lozhkov et al., 2025) and general-domain documents from FineWeb (Penedo et al., 2024) with an 8:2 token ratio. This analysis uses independently trained 16K Top-$K$ SAEs, while the main experiments use the pretrained SAE setup described in Appendix C.1. All benchmarks are evaluated with four samples per question.

Table 5: Sensitivity to SAE training budget on Qwen3.5-9B. Acc (%) and Tokens are averaged over four samples per question. Bold indicates the highest accuracy within each benchmark.

| | AIME 2026 | | HMMT 2026 | | GPQA | | HLE-STEM | |
|---|---|---|---|---|---|---|---|---|
| Training tokens | Acc | Tokens | Acc | Tokens | Acc | Tokens | Acc | Tokens |
| 0.1M | 70.00 | 18,070 | 50.00 | 17,422 | 82.83 | 17,113 | 20.69 | 15,456 |
| 1M | **77.50** | 24,366 | 55.30 | 24,270 | 83.33 | 20,597 | 20.69 | 19,329 |
| 3M | **77.50** | 25,576 | 55.30 | 26,050 | **83.84** | 23,177 | **21.84** | 21,741 |
| 10M | 75.00 | 26,351 | **56.06** | 26,539 | 78.28 | 23,655 | **21.84** | 21,156 |

Table 5 shows that SAE training data scale influences the accuracy–efficiency trade-off of Dr.CoT. With only 0.1M training tokens, the SAE achieves the shortest reasoning trajectories but lower accuracy on both mathematical benchmarks. Increasing the budget to 1M tokens improves accuracy while using one tenth of the training data of the 10M SAE, achieving 2.50% higher AIME accuracy and comparable HMMT accuracy with fewer generated tokens. However, increasing the budget from 1M to 3M does not improve mathematical accuracy but increases token consumption, suggesting diminishing returns from additional SAE training data. Overall, these results show that moderate SAE training data is sufficient to maintain the accuracy–efficiency benefits of Dr.CoT.

**Inference-Time Cost.** To quantify Dr.CoT's inference-time cost, we profile Qwen3.5-9B on AIME 2026 and decompose computation into LLM and SAE components. We measure FLOPs, decoding throughput, and peak GPU memory to characterize the computational trade-off introduced by Dr.CoT.

Table 6: Inference-time cost of Dr.CoT on Qwen3.5-9B and AIME 2026.

| | FLOPs (T) | | | | |
|---|---|---|---|---|---|
| Method | LLM | SAE | Total | Tokens/s | Peak GPU (GiB) |
| Vanilla | 544.780 | 0 | 544.780 | 31.37 | 17.78 |
| Dr.CoT | 490.674 | 15.092 | 505.766 | 30.93 | 19.78 |

Table 6 shows that Dr.CoT reduces inference computation despite the SAE overhead. Specifically, total computation decreases from 544.780 to 505.766 TFLOPs, corresponding to a 7.2% reduction. This saving introduces limited overhead: decoding throughput decreases by 1.4%, and peak GPU memory increases by 2.00 GiB. Moreover, using throughput as an estimate, Dr.CoT reduces decoding time by 9.6% on AIME runs, although this estimate does not represent end-to-end latency. Overall, Dr.CoT lowers inference computation with limited SAE memory overhead.

## 5 Conclusion

In this paper, we introduce **Dr.CoT**, a framework for diagnosing and regulating chain-of-thought reasoning through prompt-conditioned sparse feature coverage. Dr.CoT represents the prompt and evolving reasoning trajectory in a shared sparse feature space, tracks coverage dynamics for adaptive stopping, and uses uncovered prompt features to steer reasoning. Together, these components integrate reasoning-state representation, sufficiency diagnosis, and trajectory regulation within a unified feature space. Experiments show that Dr.CoT reduces average token usage by **17.01%** and improves average accuracy by **2.16%** over standard CoT, achieving a better accuracy–efficiency balance than all evaluated baselines and enabling more efficient reasoning in practice. A current limitation is that Dr.CoT derives its sparse feature representation from a single model layer. Future work could explore multi-layer representations to characterize reasoning across model depths.

### AI USE STATEMENT

Large language models were used as writing assistants to improve the clarity, conciseness, grammatical correctness, and overall readability of the manuscript, including refining sentence structure, academic phrasing, and reducing unnecessary repetition. LLMs did not contribute to research ideation, methodological development, experimental design, data analysis, interpretation of results, or the generation of the core scientific claims, arguments, or conclusions presented in this work. All research decisions, experimental results, analyses, and scientific contributions were produced and validated by the authors. The authors carefully reviewed all AI-assisted revisions to ensure technical accuracy and consistency with the intended meaning, and take full responsibility for the final content of the paper.

### REPRODUCIBILITY STATEMENT

Section 4 specifies the benchmarks, baselines, evaluation metrics, and aggregation rules, and reports the corresponding results. To facilitate reproduction, Appendix C.1 details the corresponding implementation and evaluation settings, while Appendix C.2 provides additional benchmark details. The Qwen3.5 checkpoints and pretrained Qwen-Scope SAEs used in the main experiments are also publicly available through the official Hugging Face collections (Qwen3.5 and Qwen-Scope).

## REFERENCES

Simone Caldarella, Davide Talon, Rahaf Aljundi, Elisa Ricci, and Massimiliano Mancini. Thinking past the answer: Evaluating harmful overthinking in large reasoning models. *arXiv preprint arXiv:2606.02835*, 2026.

Center for AI Safety, Scale AI, and HLE Contributors Consortium. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649:1139–1146, 2026. doi: 10.1038/s41586-025-09962-4. URL https://arxiv.org/abs/2501.14249.

Benteng Chen, Weida Wang, Shufei Zhang, Mingbao Lin, and Min Zhang. Step-grpo: Internalizing dynamic early exit for efficient reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21710–21724, 2026.

Runjin Chen, Zhenyu Zhang, Junyuan Hong, Souvik Kundu, and Zhangyang Wang. Seal: Steerable reasoning calibration of large language models for free. *arXiv preprint arXiv:2504.07986*, 2025.

Muzhi Dai, Chenxu Yang, and Qingyi Si. S-grpo: Early exit via reinforcement learning in reasoning models, 2025. URL https://arxiv.org/abs/2505.07686.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms. *arXiv preprint arXiv:2605.00674*, 2026.

Boyi Deng, Xu Wang, Yaoning Wang, Yu Wan, Yubo Ma, Baosong Yang, Haoran Wei, Jialong Tang, Huan Lin, Ruize Gao, Tianhao Li, Qian Cao, Xuancheng Ren, Xiaodong Deng, An Yang, Fei Huang, Dayiheng Liu, and Jingren Zhou. Qwen-Scope: Turning sparse features into development tools for large language models, 2026. URL https://arxiv.org/abs/2605.11887.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: Llm learns when to think. *Advances in neural information processing systems*, 2025.

Yi Fang, Wenjie Wang, Mingfeng Xue, Boyi Deng, Fengli Xu, Dayiheng Liu, and Fuli Feng. Controllable llm reasoning via sparse autoencoder-based steering. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21290–21305, 2026.

Yichao Fu, Xuewei Wang, Hao Zhang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. In *International Conference on Learning Representations*, volume 2026, pp. 94355–94377, 2026.

Jingchu Gai, Guanning Zeng, Christina Baek, Chen Wu, J Zico Kolter, Andrej Risteski, and Aditi Raghunathan. Understanding and mitigating premature confidence for better llm reasoning. *arXiv preprint arXiv:2605.24396*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645 (8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Zirui He, Haiyan Zhao, Yiran Qiao, Fan Yang, Ali Payani, Jing Ma, and Mengnan Du. Saif: A sparse autoencoder framework for interpreting and steering instruction following of language models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 180–195. Springer, 2026.

Yao Huang, Huanran Chen, Shouwei Ruan, Yichi Zhang, Xingxing Wei, and Yinpeng Dong. Mitigating overthinking in large reasoning models via manifold steering, 2025. URL https://arxiv.org/abs/2505.22411.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024.

Shuyang Jiang, Yusheng Liao, Ya Zhang, Yanfeng Wang, and Yu Wang. Overthinking reduction with decoupled rewards and curriculum data scheduling. In *International Conference on Learning Representations*, volume 2026, pp. 114966–114995, 2026.

Junyeob Kim, Sang-goo Lee, and Taeuk Kim. Think just enough: Leveraging self-assessed confidence for adaptive reasoning in language models. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 5000–5006, 2026.

Yulin Li, Tengyao Tu, Li Ding, Junjie Wang, Huiling Zhen, Yixin Chen, Yong Li, and Zhuotao Tian. Efficient reasoning with balanced thinking. *arXiv preprint arXiv:2603.12372*, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Zhengkai Lin, Zhihang Fu, Ze Chen, Chao Chen, Liang Xie, Wenxiao Wang, Deng Cai, Zheng Wang, and Jieping Ye. Controlling thinking speed in reasoning models. *arXiv preprint arXiv:2507.03704*, 2025.

Anton Lozhkov, Hynek Kydlíček, Loubna Ben Allal, Guilherme Penedo, Edward Beeching, Quentin Gallouédec, Nathan Habib, Lewis Tunstall, and Leandro von Werra. Openr1-math-220k. https://huggingface.co/datasets/open-r1/OpenR1-Math-220k, 2025.

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.

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.

Samuel Marks, Can Rager, Eric Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *International Conference on Learning Representations*, volume 2025, pp. 23888–23923, 2025.

Dehai Min, Giovanni Vaccarino, Huiyi Chen, Yongliang Wu, Gal Yona, and Lu Cheng. Stop when reasoning converges: Semantic-preserving early exit for reasoning models. *arXiv preprint arXiv:2605.17672*, 2026.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL https://arxiv.org/abs/2406.17557.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. Dast: Difficulty-adaptive slow-thinking for large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 2322–2331, 2025.

Renliang Sun, Wei Cheng, Dawei Li, Haifeng Chen, and Wei Wang. Stop when enough: Adaptive early-stopping for chain-of-thought reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27250–27268, 2026.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Chenlong Wang, Yuanning Feng, Dongping Chen, Zhaoyang Chu, Ranjay Krishna, and Tianyi Zhou. Wait, we don't need to "wait"! removing thinking tokens improves reasoning efficiency. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 7459–7482, Suzhou, China,

November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.394. URL https://aclanthology.org/2025.findings-emnlp.394/.

Jiahao Wang, Bingyu Liang, Chenhao Hu, Longhui Zhang, Xuebo Liu, Min zhang, Jing Li, and Xuelong Li. Suco: Sufficiency-guided continuous adaptive reasoning, 2026. URL https://arxiv.org/abs/2606.17687.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Hao Wen, Xinrui Wu, Yi Sun, Feifei Zhang, Liye Chen, Jie Wang, Yunxin Liu, Yunhao Liu, Ya-Qin Zhang, and Yuanchun Li. Budgetthinker: Empowering budget-aware llm reasoning with control tokens. *arXiv preprint arXiv:2508.17196*, 2025.

Menghua Wu, Cai Zhou, Stephen Bates, and Tommi Jaakkola. Thought calibration: Efficient and confident test-time scaling. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14291–14305, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.722. URL https://aclanthology.org/2025.emnlp-main.722/.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. Tokenskip: Controllable chain-of-thought compression in llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3351–3363, 2025.

Yang Xiang, Yixin Ji, Ruotao Xu, Dan Qiao, Zheming Yang, Juntao Li, and Min Zhang. When is thinking enough? early exit via sufficiency assessment for efficient reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23541–23556, 2026.

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

Ting Xu, Xu He, Yupu Lu, Jiankai Sun, Dong Li, Wai Lam, and Jianye Hao. Unveiling the entropy dynamics of chain-of-thought reasoning. *arXiv preprint arXiv:2606.02020*, 2026.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weipinng Wang. Dynamic early exit in reasoning models. In *International Conference on Learning Representations*, volume 2026, pp. 88170–88210, 2026.

Wenkai Yang, Shuming Ma, Yankai Lin, and Furu Wei. Towards thinking-optimal scaling of test-time compute for llm reasoning. *arXiv preprint arXiv:2502.18080*, 2025.

Weiqi Zhai, Zhihai Wang, Jinghang Wang, Boyu Yang, Xiaogang Li, Xander Xu, Bohan Wang, Peng Wang, Xingzhe Wu, Anfeng Li, Qiyuan Feng, Yuhao Zhou, Taolin Han, Wenjie Luo, Yiyuan Li, Xiang Zheng, Yaxuan Wang, Ruixiang Luo, Guojie Lin, Peiyao Xiao, Chengliang Xu, Ben Wang, Zeyu Wang, Zichao Chen, Jianan Ye, Yijie Hu, Jialong Chen, Zongwen Shen, Yuliang Xu, An Yang, Bowen Yu, Dayiheng Liu, Junyang Lin, Hu Wei, Que Shen, and Bing Zhao. Hle-verified: A systematic verification and structured revision of humanity's last exam, 2026. URL https://arxiv.org/abs/2602.13964.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3716–3730, 2025.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2026, 2026.

# A PRELIMINARIES: SPARSE AUTOENCODERS

**Architecture.** At a selected transformer layer $\ell$, a Top-$K$ sparse autoencoder (SAE) (Deng et al., 2026) encodes a residual-stream activation $h \in \mathbb{R}^d$ into sparse features:

$$z(h) = [\mathrm{TopK}_K(W_{\mathrm{enc}}h + b_{\mathrm{enc}})]_+ . \tag{11}$$

Here, $W_{\mathrm{enc}} \in \mathbb{R}^{M\times d}$ and $M$ denotes the number of SAE features. The Top-$K$ operator retains the $K$ largest encoder pre-activations, while $[\cdot]_+$ sets negative values to zero, yielding at most $K$ active features per token. The decoder then reconstructs the residual-stream activation as

$$\widehat{h} = \mathrm{Dec}(z(h)) = W_{\mathrm{dec}}z(h) + b_{\mathrm{dec}}, \tag{12}$$

Here, $W_{\mathrm{dec}} \in \mathbb{R}^{d\times M}$ denotes the decoder matrix. Together, the encoder and decoder define a sparse feature space and a corresponding mapping from feature differences back to the residual stream.

**SAE Training Objective.** A Top-$K$ SAE is trained to reconstruct residual-stream activations from sparse feature representations. For activations drawn from a training distribution $\mathcal{D}_h$, a standard reconstruction objective is

$$\mathcal{L}_{\mathrm{rec}} = \mathbb{E}_{h\sim\mathcal{D}_h}\left[\|h - \mathrm{Dec}(z(h))\|_2^2\right] . \tag{13}$$

Together with Top-$K$ sparsification, this objective encourages the learned sparse features to preserve information needed to reconstruct the original residual-stream activations.

# B ALGORITHMIC DETAILS

## B.1 DR.COT ALGORITHM

Algorithm 1 summarizes the core procedure of Dr.CoT, comprising prompt-conditioned feature modeling, coverage-guided sufficiency diagnosis, and residual feature steering. Dr.CoT computes the prompt feature state $F_x$ and incrementally accumulates the reasoning state $G_t$ during generation. At each diagnosis step, $c_t$ and $g_t$ determine reasoning sufficiency; otherwise, the uncovered feature residual $R_t$ is decoded into a residual-stream steering direction for continued reasoning.

**Algorithm 1** Dr.CoT algorithm

**Input:** prompt $x$; frozen language model and SAE; target layer $\ell$
**Parameters:** coverage threshold $\tau_c$, gain threshold $\tau_g$, window size $W$, patience $P$, and steering strength $\lambda$
$F_x \leftarrow \sum_i z(h_i^x)$; $G_0 \leftarrow 0$
**for** each reasoning token $t$ **do**
  *(a) Prompt-Conditioned Feature Modeling*
  $G_t \leftarrow G_{t-1} + z(h_t)$; $R_t \leftarrow [F_x - G_t]_+$
  *(b) Coverage-Guided Sufficiency Diagnosis*
  $c_t \leftarrow 1 - \dfrac{\|R_t\|_1}{\|F_x\|_1}$
  **if** $t \bmod W = 0$ **then**
    $g_t \leftarrow \dfrac{c_t - c_{t-W}}{W}$
    **if** $c_t \geq \tau_c$ and $\max_{0\leq k<P} g_{t-kW} < \tau_g$ **then**
      **stop reasoning**
    **end if**
  **end if**
  *(c) Residual Feature Steering*
  $v_t \leftarrow W_{\mathrm{dec}}R_t + b_{\mathrm{dec}}$
  Apply residual feature steering with direction $v_t$ and strength $\lambda$
**end for**
**return** Generated response

## B.2 MATHEMATICAL FORMULATION OF PROMPT-CONDITIONED FEATURE COVERAGE

Given a prompt $x$ and its evolving reasoning prefix $y_{1:t}$, we seek to quantify how much of the prompt representation has been covered by the reasoning trajectory. This relation is asymmetric: the prompt

provides a fixed coverage reference, whereas the reasoning trajectory may introduce additional features needed to solve the problem. Accordingly, coverage is measured relative to the prompt rather than symmetrically between the two states.

**Optimal covered component.** Let $F_x \in \mathbb{R}^M_{\geq 0}$ denote the prompt feature state and $G_t \in \mathbb{R}^M_{\geq 0}$ the accumulated reasoning state, as defined in Equation 3. We define the covered component through the following constrained maximization:

$$C_t^\star = \arg\max_{C \in \mathbb{R}^M_{\geq 0}} \|C\|_1 \quad \text{s.t.} \quad C \preceq F_x,\ C \preceq G_t, \tag{14}$$

where $\preceq$ denotes coordinate-wise inequality. Since the optimization is separable across feature dimensions, its unique solution is

$$C_t^\star = \min(F_x, G_t), \tag{15}$$

where the minimum is taken coordinate-wise. Thus, $C_t^\star$ gives the feature-wise overlap between the prompt and accumulated reasoning activations.

**Covered–uncovered decomposition.** The complementary uncovered feature residual is therefore

$$R_t = F_x - C_t^\star = [F_x - G_t]_+, \tag{16}$$

yielding the decomposition as follows:

$$F_x = C_t^\star + R_t. \tag{17}$$

By construction, each feature contributes at most its corresponding prompt activation. Feature activation beyond its prompt representation contributes no additional coverage, so excess activation in one feature cannot compensate for uncovered prompt features elsewhere.

**Normalized coverage.** For $\|F_x\|_1 > 0$, we define the coverage ratio as the fraction of prompt feature activation contained in the covered component:

$$c_t = \frac{\|C_t^\star\|_1}{\|F_x\|_1} = 1 - \frac{\|R_t\|_1}{\|F_x\|_1}. \tag{18}$$

Thus, $c_t$ measures the fraction of prompt feature activation covered by the reasoning trajectory.

**Coverage properties.** Since $0 \preceq C_t^\star \preceq F_x$, the coverage ratio is bounded as

$$0 \leq c_t \leq 1. \tag{19}$$

Moreover, the nonnegativity of SAE activations implies

$$G_{t+1} = G_t + z(h_{t+1}) \succeq G_t. \tag{20}$$

Since the coordinate-wise minimum is monotone in its second argument,

$$C_{t+1}^\star = \min(F_x, G_{t+1}) \succeq \min(F_x, G_t) = C_t^\star, \tag{21}$$

and therefore

$$c_{t+1} \geq c_t. \tag{22}$$

Finally, full prompt-feature coverage is attained exactly when the uncovered residual vanishes:

$$c_t = 1 \iff R_t = 0 \iff G_t \succeq F_x. \tag{23}$$

Hence, $c_t$ provides a bounded and non-decreasing measure of prompt-conditioned reasoning progress.

# C Experimental Details

## C.1 Implementation and Evaluation Settings

**Backbones and SAE configuration.** We evaluate Qwen3.5-2B, Qwen3.5-9B, and Qwen3.5-27B (Qwen Team, 2026). For sparse feature extraction and residual feature steering, we use transformer layers 20, 28, and 60, respectively, with pretrained Top-$K$ SAEs ($K = 100$) (Deng et al.,

2026). Both the LLM backbone and SAE are kept frozen throughout inference. The Dr.CoT hyperparameter settings for each model are summarized in Table 7. For each backbone, we calibrate hyperparameters once on 50 random MATH-500 problems (Lightman et al., 2024) and fix them across benchmarks. Across all benchmarks, we adopt the Qwen chat template and the same zero-shot evaluation prompt, as specified below. For decoding, we set the temperature to 0.6, top-$p$ to 0.95, and top-$k$ to 20, cap generation at 32,768 tokens, and generate four independent responses per question.

**Evaluation Prompt**

**System prompt:** `"Please reason step by step, and put your final answer within \boxed{}."`

**Few-shot prompt:** `""`

**Question format:** `"{question}"`

Table 7: Core hyperparameters of Dr.CoT.

| Parameter | 2B | 9B | 27B |
|---|---|---|---|
| Target layer $\ell$ | 20 | 28 | 60 |
| Coverage threshold $\tau_c$ | 0.8 | 0.8 | 0.8 |
| Coverage window $W$ | 128 | 128 | 128 |
| Coverage gain threshold $\tau_g$ | $1 \times 10^{-6}$ | $5 \times 10^{-8}$ | $5 \times 10^{-8}$ |
| Patience $P$ | 3 | 3 | 3 |
| Steering strength $\lambda$ | 0.01 | 0.01 | 0.01 |

**Baselines** We compare Dr.CoT with Vanilla, NoThinking, and three representative adaptive early-stopping methods: DEER (Yang et al., 2026), REFRAIN (Sun et al., 2026), and CUSUM (Xu et al., 2026). We follow the original papers and official implementations when available, while keeping the common decoding settings. Vanilla uses standard CoT reasoning, whereas NoThinking disables the thinking mode via `enable_thinking=False`. DEER greedily probes up to 20 answer tokens at each `Wait` transition and stops when the probe reaches `</think>` with confidence above 0.95, using a reasoning-budget fraction of 0.8 and at most 10 probes. REFRAIN detects reflective repetition using `all-MiniLM-L6-v2` embeddings, with the stopping threshold selected by a sliding-window Upper Confidence Bound (UCB) strategy from $\{0.60, 0.65, 0.70, 0.75, 0.80\}$. CUSUM uses a decision threshold of 76.4302 and performs greedy probes every 128 tokens, with each probe capped at 128 generated tokens.

**Evaluation protocol and token counting.** All methods use the same boxed-answer extraction and grading procedure. The last boxed expression is taken as the prediction and normalized before multiple-choice matching, numerical equivalence, or symbolic equivalence checking, as appropriate. Responses without a boxed answer are marked incorrect. Accuracy is averaged over all sampled responses. Token counts include reasoning, final-answer, answer-transition, and method-specific probe tokens. Overall denotes the macro-average across the four benchmarks. For cross-backbone summaries, we average Overall accuracy and relative token reduction over Vanilla across backbones.

**Implementation details.** Experiments are implemented using Hugging Face Transformers with KV caching enabled for autoregressive decoding. Backbone inference is performed in bfloat16, while SAE-related computations are carried out in float32 throughout inference. We use PyTorch 2.10.0 and Transformers 5.12.1 for all experiments.

### C.2 Evaluation Benchmarks

As shown in Table 8, we evaluate four benchmarks across mathematics and science domains.

**AIME 2026.** The 2026 American Invitational Mathematics Examination (AIME) (Zhang & Math-AI, 2026) consists of two 15-question contests, AIME I and AIME II, yielding 30 competition-mathematics problems in total. We use the `math-ai/aime26` release, retaining the original problem statements and reference answers.

Table 8: Overview of the evaluation benchmarks.

| Benchmark | Domain | Questions | Answer format |
|---|---|---|---|
| AIME 2026 | Competition mathematics | 30 | Integer |
| HMMT February 2026 | Competition mathematics | 33 | Mathematical answer |
| GPQA-Diamond | Graduate-level science | 198 | Multiple choice (4 options) |
| HLE-STEM | Expert-level STEM | 87 | Multiple choice |

**HMMT February 2026.** The February 2026 Harvard–MIT Mathematics Tournament (HMMT) (Dekoninck et al., 2026) is a competition-mathematics benchmark drawn from `MathArena/hmmt_feb_2026`. We evaluate 33 problems, comprising all 30 individual-round problems and Problems 2, 3, and 5 from the Team Round.

**GPQA-Diamond.** The Graduate-Level Google-Proof Q&A (GPQA) Diamond benchmark (Rein et al., 2023) consists of 198 graduate-level multiple-choice science questions with four answer options. We evaluate exact-match accuracy by comparing the final boxed option against the reference answer.

**HLE-STEM.** Humanity's Last Exam (HLE) (Center for AI Safety et al., 2026) provides the source data for HLE-STEM, which we construct from its verified subset, HLE-Verified (Zhai et al., 2026). From the 2,500 source examples, we retain 173 Gold multiple-choice items with no associated images. We further restrict this set to non-mathematical STEM subjects. The final subset contains 87 text-only questions from biology, medicine, chemistry, computer science, artificial intelligence, engineering, and physics. We evaluate exact-match accuracy by comparing the final uppercase option letter with the corresponding reference answer.

# D Case Studies

In this section, we present four representative Qwen3.5-27B cases, one from each benchmark, to examine how Dr.CoT operates in practice. We analyze each case from two complementary perspectives. First, we examine the prompt-conditioned coverage dynamics to characterize how reasoning sufficiency evolves throughout generation. Second, we compare the corresponding reasoning trajectories with Vanilla to illustrate how Dr.CoT regulates the reasoning process and shapes the final outcomes. Together, these analyses connect Dr.CoT's internal coverage dynamics with their observable effects on reasoning behavior. All cases use the same full Dr.CoT configuration as in the main experiments.

## D.1 Prompt-Conditioned Coverage Dynamics

To examine how Dr.CoT assesses reasoning sufficiency during generation, we track the evolution of prompt-conditioned coverage $c_t$ and coverage gain $g_t$ throughout the four selected cases. Figure 3 visualizes these dynamics together with the corresponding sufficiency thresholds and stopping points. All cases use the same coverage and stopping configuration as in the main experiments.

As shown in Figure 3, all four cases follow a consistent pattern under Coverage-Guided Sufficiency Diagnosis. The coverage ratio $c_t$ exceeds the coverage threshold $\tau_c$ in every case and subsequently increases by an additional 0.035 to 0.082 before termination. This continued increase indicates that crossing the coverage threshold alone does not establish reasoning sufficiency, since subsequent reasoning still contributes additional prompt-feature coverage. Dr.CoT therefore jointly considers the coverage ratio and coverage gain, terminating reasoning only when $c_t \geq \tau_c$ and $g_t$ remains below the coverage-gain threshold $\tau_g$ for $P = 3$ consecutive windows. The former ensures sufficient accumulated prompt-feature coverage, whereas the latter indicates that continued reasoning contributes little additional coverage. Across the four cases, Dr.CoT reduces output length by 12.9% to 58.8% relative to Vanilla while preserving the correct final answer in each case.

## D.2 Reasoning Trajectory Comparisons

We next examine four selected cases, one from each benchmark, to characterize how Dr.CoT regulates the reasoning trajectory. All cases use the same full Dr.CoT configuration as in the main evaluation.

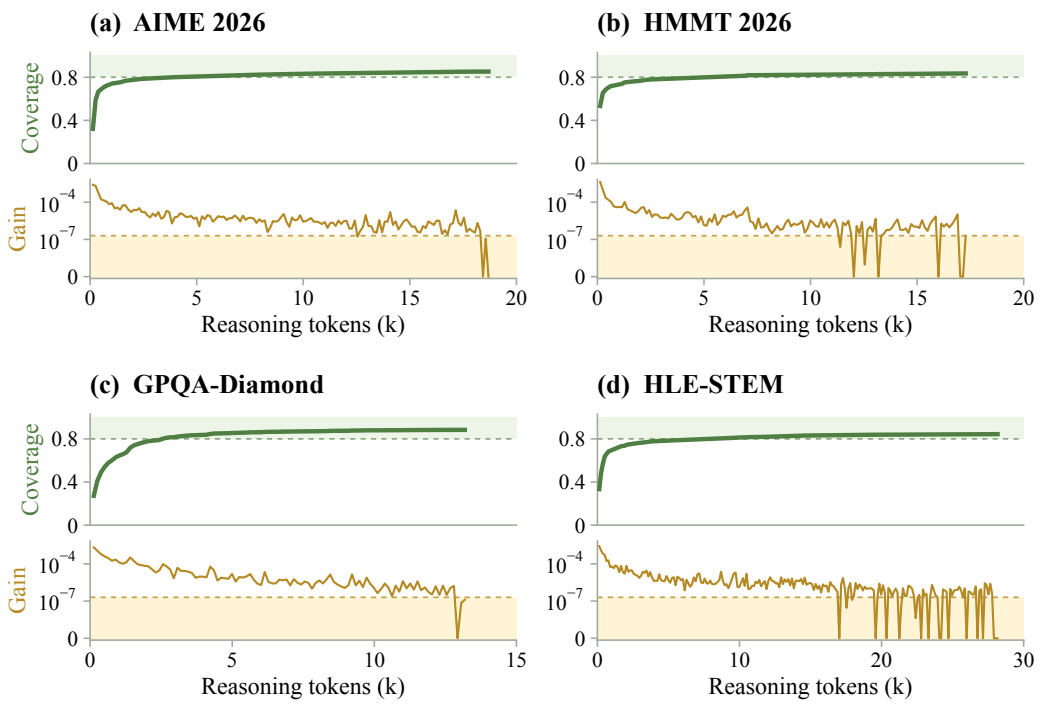


Figure 3: **Prompt-conditioned coverage dynamics.** Green and gold curves show the coverage ratio $c_t$ and coverage gain $g_t$, respectively. Dashed lines mark $\tau_c$ and $\tau_g$, with shaded regions indicating threshold-satisfying regions. Each trajectory terminates when the stopping criterion is satisfied.

As shown in Table 9, Dr.CoT is the only method among the six that solves all four selected cases, while reducing generated tokens by 12.9% to 58.8% relative to Vanilla. The trajectory comparisons below further illustrate how Coverage-Guided Sufficiency Diagnosis determines reasoning sufficiency, while Residual Feature Steering guides continued reasoning using uncovered prompt features.

Table 9: Case-level outcomes across four benchmarks. The table reports final-answer correctness and generated-token counts for all six methods on the selected cases.

| Method | AIME 2026 | | HMMT 2026 | | GPQA-Diamond | | HLE-STEM | |
|---|---|---|---|---|---|---|---|---|
| | Solved | Tokens | Solved | Tokens | Solved | Tokens | Solved | Tokens |
| Vanilla | ✓ | 32,644 | ✓ | 32,647 | × | 32,642 | × | 32,642 |
| NoThinking | × | 8,067 | × | 8,066 | ✓ | 1,739 | ✓ | 1,420 |
| DEER | ✓ | 26,427 | ✓ | 19,566 | × | 21,037 | × | 26,395 |
| REFRAIN | ✓ | 18,148 | ✓ | 23,232 | × | 5,605 | × | 6,614 |
| CUSUM | × | 28,519 | ✓ | 32,372 | × | 33,150 | × | 15,614 |
| Dr.CoT | ✓ | 18,957 | ✓ | 17,553 | ✓ | 13,451 | ✓ | 28,427 |

The following comparisons show how Coverage-Guided Sufficiency Diagnosis identifies when reasoning is sufficient, while Residual Feature Steering guides subsequent reasoning using uncovered prompt features. For each case, we present side-by-side excerpts from Vanilla and Dr.CoT, followed by a brief analysis of the key reasoning differences and their effects on correctness and generated-token counts. Step indices indicate excerpt order only. Vertical ellipses indicate omitted reasoning between excerpts, while inline ellipses indicate omitted text within an excerpt. Step headings are added for readability, with only formatting normalized.

### Case 1: AIME 2026

**Problem.** Count positive-integer sequences summing to 12 whose left-to-right evaluation is zero, where $a \circ b = a - b$ if $a$ is odd and $b$ is even, and $a \circ b = a + b$ otherwise.

| Vanilla | 32,644 tokens |
|---|---|
| ⋮ | |
| **Step $i$: Obtain the count.** | |
| Total overall: $48 + 48 + 61 = 157$. | |
| **Step $j$: Test a smaller sum.** | |
| For sum=4, target should be 2. … So $1, 2, 1$ is a solution. | |
| ⋮ | |
| **Step $k$: Find a mismatch.** | |
| Sum contribution: $1 \cdot 1 = 1$. We need sum = 2. So $1 \neq 2$. | |
| **Step $\ell$: Recover the factor.** | |
| For sum=4, target is 2. Here we got $2 \times 1 = 2$. So the formula works! Why did I think it gave 1? | |
| ⋮ | |
| **Step $N$: Final answer** | |
| $\boxed{157}$ (✔) | |

| Dr.CoT | 18,957 tokens |
|---|---|
| ⋮ | |
| **Step $i$: Balance signed sums.** | |
| Since the total sum is 12, each sum must be 6. | |
| **Step $j$: Form parity blocks.** | |
| Let the sequence of odd numbers be $o_1, o_2, \ldots, o_m$. … Block 0: even numbers before $o_1$. Sign $+$. Block 1: even numbers between $o_1$ and $o_2$. Sign $-$. | |
| ⋮ | |
| **Step $k$: Count each case.** | |
| Total for $m = 2$: $20 + 16 + 12 = 48$. … Total for $m = 4$: $36 + 48 = 84$. … Total for $m = 6$: 25. | |
| **Step $\ell$: Combine the counts.** | |
| Total number of sequences = $48 + 84 + 25 = 157$. | |
| ⋮ | |
| **Step $N$: Final answer** | |
| $\boxed{157}$ (✔) | |

**Case 1 analysis.** Both methods obtain the correct count of 157, but Vanilla re-evaluates a smaller instance, encounters a factor-of-two inconsistency, and revisits the derivation. Dr.CoT instead finalizes after combining the case counts, preserving the correct answer with 41.9% fewer generated tokens.

### Case 2: HMMT 2026

**Problem.** In a triangle with $AB = 5$, $AC = 7$, and incenter $I$, the incircle $\omega$ intersects the circumcircle of $BIC$ at $E, F$. A distinct circle $\Gamma$ through $E, F$, tangent to lines $AB$, $AC$, has twice the incircle radius. Find $a = BC$. Here $u = (12 - a)/2$, $b = 7$, $c = 5$, and $\alpha = A/2$.

| Vanilla | 32,647 tokens |
|---|---|
| ⋮ | |
| **Step $i$: Derive the quadratic.** | |
| $2u^2 - 18u + 35 = 0$. | |
| **Step $j$: Choose the valid side.** | |
| So $3 - \sqrt{11} < 0$. … $a = 3 + \sqrt{11}$. | |
| ⋮ | |
| **Step $k$: Reopen the relation.** | |
| $18u - 3u^2 = (35 - u^2)\cos\alpha$ | |
| **Step $\ell$: Encounter a degeneracy.** | |
| So we must have $\cos\alpha = 1$. This implies $A = 0$. … Let's re-evaluate the radical axis equation. | |
| ⋮ | |
| **Step $N$: Final answer** | |
| $\boxed{3 + \sqrt{11}}$ (✔) | |

| Dr.CoT | 17,553 tokens |
|---|---|
| ⋮ | |
| **Step $i$: Derive the quadratic.** | |
| $2u^2 - 18u + 35 = 0$. | |
| **Step $j$: Choose the valid side.** | |
| So $3 - \sqrt{11} < 0$. … $a = 3 + \sqrt{11}$. | |
| ⋮ | |
| **Step $k$: Check circle incidence.** | |
| $E$ and $F$ have the same power wrt $\omega$ and $\Gamma$. … So $E, F$ lie on $\Gamma$. | |
| **Step $\ell$: Check the side length.** | |
| Check triangle inequality: $a < b + c = 12$. … $a > \lvert b - c \rvert = 2$. | |
| ⋮ | |
| **Step $N$: Final answer** | |
| $\boxed{3 + \sqrt{11}}$ (✔) | |

**Case 2 analysis.** Both trajectories derive $a = 3 + \sqrt{11}$ as the valid side length. Vanilla then reopens the radical-axis relation, introducing an extra cosine factor that leads to an apparent degeneracy. Dr.CoT avoids this detour and returns the correct answer with 46.2% fewer generated tokens.

**Case 3: GPQA-Diamond**

**Problem.** Count stars detectable by ESPRESSO at Paranal (one 8 m VLT telescope; one hour; S/N $\geq 10$ per binned pixel). Contrast Polaris (star b) and star c ($V = 15$). Reference: A (three stars).

| Vanilla — 32,642 tokens | Dr.CoT — 13,451 tokens |
|---|---|
| ⋮ | ⋮ |
| **Step $i$: Use brightness.** | **Step $i$: Apply site visibility.** |
| Polaris is V=2. S/N will be huge. | It never rises above the horizon at Paranal. |
| **Step $j$: Include Polaris.** | **Step $j$: State the binning assumption.** |
| So count is 4. Stars: a (11), b (2), c (15), e (13.25). | If "per pixel" in spec means detector pixel. And "per binned pixel" means binned to resolution element. Resolution element is $\sim 3$ pixels. |
| ⋮ | ⋮ |
| **Step $k$: Revisit the faint limit.** | **Step $k$: Account for binning.** |
| If c is out, count is 3. But V=15 is well within the V=17 limit. So c is in. | So if we account for binning (3 pixels), the limit is V=15.2. |
| **Step $\ell$: Recheck brightness.** | **Step $\ell$: Count the visible stars.** |
| Let's check the magnitude of Star b again. Polaris. V=2.0. Detectable. So 4 stars. | In this case, Star c (V=15) is detectable ($15 < 15.2$). So count = 3. |
| ⋮ | ⋮ |
| **Step $N$: Final answer** | **Step $N$: Final answer** |
| $\boxed{C}$ (✗) | $\boxed{A}$ (✓) |

**Case 3 analysis.** Dr.CoT consistently applies the prompt-relevant visibility and binning constraints throughout the reasoning process, excluding Polaris and retaining star c to obtain A. Vanilla instead includes Polaris based on brightness, overlooking the site-visibility constraint, and returns C.

**Case 4: HLE-STEM**

**Problem.** Which introduced invasive species has the largest negative ecosystem impact in New Mexico? Compare A: *Apis mellifera* and E: *Leptinotarsa decemlineata*. The reference answer is A.

| Vanilla — 32,642 tokens | Dr.CoT — 28,427 tokens |
|---|---|
| ⋮ | ⋮ |
| **Step $i$: Apply the species constraint.** | **Step $i$: Apply the species constraint.** |
| E is native. A is introduced. | E is native. A is introduced. |
| **Step $j$: Assess the introduced species.** | **Step $j$: Assess the introduced species.** |
| So A is a better candidate for "introduced". … It is a dominant pollinator. It can outcompete native bees. | So A is a better candidate for "introduced". … It is a dominant pollinator. It can outcompete native bees. |
| ⋮ | ⋮ |
| **Step $k$: Reconsider an unverified claim.** | **Step $k$: Retain the candidate.** |
| Wait, I found a reference to **Leptinotarsa decemlineata** … Wait, I am not finding it. | Okay, I will choose **A**. |
| ⋮ | ⋮ |
| **Step $N$: Final answer** | **Step $N$: Final answer** |
| $\boxed{E}$ (✗) | $\boxed{A}$ (✓) |

**Case 4 analysis.** Both trajectories initially identify A as introduced and E as native. Dr.CoT preserves this distinction and returns A, whereas Vanilla later reverses its conclusion and outputs E. This case shows how continued generation can overturn an already correct candidate after the relevant constraint has been established, introducing unnecessary uncertainty into the final decision.

# E HYPERPARAMETER SENSITIVITY

We examine the sensitivity of Dr.CoT to the coverage threshold $\tau_c$, coverage gain threshold $\tau_g$, and steering strength $\lambda$ on Qwen3.5-9B and AIME 2026. Table 10 reports three five-point sweeps, varying one parameter at a time around the default values $\tau_c = 0.8$, $\tau_g = 5 \times 10^{-8}$, and $\lambda = 0.01$, while keeping all remaining settings fixed. Following the main evaluation, results are averaged over four samples per question on the same 30 questions.

Table 10: Hyperparameter sensitivity of Dr.CoT on Qwen3.5-9B and AIME 2026. Each block varies one hyperparameter while fixing the others at their default values. Acc (%) and Tokens are averaged over four samples per question. Tokens count all generated tokens, including both reasoning and final-answer generation. Here, † denotes the default configuration used in the main evaluation.

| (a) Coverage threshold | | | (b) Coverage gain threshold | | | (c) Steering strength | | |
|---|---|---|---|---|---|---|---|---|
| $\tau_c$ | Acc (%) | Tokens | $\tau_g$ | Acc (%) | Tokens | $\lambda$ | Acc (%) | Tokens |
| 0.6 | 83.33 | 25,384 | $2\times10^{-9}$ | 76.67 | 28,606 | 0 | 77.50 | 26,982 |
| 0.7 | 80.83 | 27,060 | $10^{-8}$ | 80.00 | 27,300 | 0.005 | 83.33 | 26,445 |
| $0.8^{\dagger}$ | 81.67 | 27,247 | $5\times10^{-8\dagger}$ | 81.67 | 27,247 | $0.01^{\dagger}$ | 81.67 | 27,247 |
| 0.9 | 80.00 | 29,349 | $2.5\times10^{-7}$ | 75.00 | 23,019 | 0.02 | 80.00 | 27,838 |
| 1.0 | 80.00 | 30,154 | $1.25\times10^{-6}$ | 60.00 | 12,069 | 0.04 | 78.33 | 28,427 |

## E.1 SENSITIVITY TO SUFFICIENCY THRESHOLDS

**Coverage threshold.** The coverage threshold $\tau_c$ specifies the minimum prompt-feature coverage required before reasoning can be diagnosed as sufficient. As shown in Table 10(a), accuracy varies only from 80.00% to 83.33% as $\tau_c$ increases from 0.6 to 1.0, while generated tokens increase from 25,384 to 30,154. This trend indicates that $\tau_c$ primarily controls reasoning length by requiring greater prompt-feature coverage before stopping, with limited effect on accuracy over the tested range. The default $\tau_c = 0.8$ lies within this stable regime, indicating limited sensitivity to its precise value.

**Coverage gain threshold.** The coverage gain threshold $\tau_g$ specifies the maximum coverage gain allowed over the $P$ consecutive windows used to diagnose reasoning sufficiency. As shown in Table 10(b), varying $\tau_g$ from $10^{-8}$ to $5 \times 10^{-8}$ near the default has little impact on accuracy or generated-token counts. A smaller $\tau_g$ imposes a stricter stopping condition, increasing generated-token counts without improving accuracy, whereas larger values permit earlier stopping at the cost of progressively lower accuracy. The default $\tau_g = 5 \times 10^{-8}$ therefore provides a practical balance between reasoning efficiency and answer accuracy within the tested range.

## E.2 SENSITIVITY TO STEERING STRENGTH

The steering strength $\lambda$ controls the magnitude of the residual-stream intervention derived from uncovered prompt features. As shown in Table 10(c), moderate nonzero values of $\lambda$ from 0.005 to 0.02 achieve 80.00%–83.33% accuracy, consistently exceeding the 77.50% obtained without steering, while generated-token counts vary only moderately. Increasing $\lambda$ to 0.04 reduces accuracy to 78.33%, indicating that overly strong steering can degrade reasoning performance. The default $\lambda = 0.01$ therefore represents a moderate intervention: strong enough to improve over no steering, yet restrained enough to avoid the accuracy loss observed at larger values.

Overall, Dr.CoT remains robust to moderate hyperparameter variations around the default configuration. Across the tested ranges, $\tau_c$ controls reasoning length, $\tau_g$ governs stopping aggressiveness, and $\lambda$ regulates steering strength, with the defaults achieving a favorable accuracy–efficiency trade-off.