

S³-CoT: Self-Sampled Succinct Reasoning Enables Efficient Chain-of-Thought LLMs

Yanrui Du¹, Sendong Zhao^{1*}, Yibo Gao¹, Danyang Zhao¹, Qika Lin¹, Ming Ma¹,
Jiayun Li¹, Yi Jiang¹, Kai He², Qianyi Xu², Bing Qin¹, Mengling Feng²

¹Harbin Institute of Technology, Harbin, China

²National University of Singapore, Singapore

{yrdu,sdzhao}@ir.hit.edu.cn

Abstract

Large language models (LLMs) equipped with chain-of-thought (CoT) achieve strong performance and offer a window into LLM behavior. However, recent evidence suggests that improvements in CoT capabilities often come with redundant reasoning processes, motivating a key question: can LLMs acquire a “fast-thinking” mode analogous to human System 1 reasoning? To explore this, our study presents a self-sampling framework based on activation steering for efficient CoT learning. Our method can induce style-aligned and variable-length reasoning traces from target LLMs themselves without any teacher guidance, thereby alleviating a central bottleneck of SFT-based methods—the scarcity of high-quality supervision data. Using filtered data by gold answers, we perform SFT for efficient CoT learning with (i) a human-like dual-cognitive system, and (ii) a progressive compression curriculum. Furthermore, we explore a self-evolution regime in which SFT is driven solely by prediction-consistent data of variable-length variants, eliminating the need for gold answers. Extensive experiments on math benchmarks, together with cross-domain generalization tests in medicine, show that our method yields stable improvements for both general and R1-style LLMs. Our data and model checkpoints can be found at <https://github.com/DYR1/S3-CoT>.

1 Introduction

Chain-of-thought (CoT) has become a standard mechanism for eliciting multi-step reasoning in large language models (LLMs), substantially improving performance on complex tasks (Wei et al., 2022; Yao et al., 2023; Besta et al., 2024; Wang et al., 2022). More recently, the field has shifted toward internalizing such reasoning behaviors into LLMs themselves via post-training pipelines, aiming to make strong reasoning the default rather than

prompt-contingent (Zhao et al., 2024; Jaech et al., 2024; Guo et al., 2025; Yu et al., 2024). However, once reasoning is internalized, the generated reasoning traces often become overly long and redundant, inflating latency and cost even on easy instances (Wu et al., 2025; Liu et al., 2025). This motivates methods that *compress reasoning traces while preserving reasoning ability*.

To achieve this, existing work can be grouped into three categories. (1) **Prompt-based control** constrains reasoning length via explicit budgets or specialized templates (Han et al., 2025; Nayab et al., 2024). While lightweight, these methods are highly sensitive to prompt wording, and often require task- or model-specific tuning. (2) **SFT-based methods** fine-tune LLMs with curated concise traces as supervision (Munkhbat et al., 2025). Their primary bottleneck is supervision data: collecting high-quality and variable-length CoTs is expensive and difficult. Both C3oT (Kang et al., 2025) and CoT-Valve (Ma et al., 2025) require guidance from external tools or teacher LLMs to achieve this. Such teacher-dependent pipelines can be brittle, as CoT verbosity and stylistic conventions vary widely across LLM families. (3) **RL-based methods** explicitly optimize the length–accuracy trade-off by shaping rewards or enforcing token constraints during training (Hou et al., 2025; Liu et al., 2025; Tu et al., 2025). Although effective, RL typically incurs a higher computational cost and is sensitive to reward design and training stability.

In our study, we target **the gap left by SFT-based methods**: how to obtain high-quality, style-aligned, variable-length CoT data without any teacher guidance. Inspired by activation steering, we propose S³-CoT, a self-sampling framework for efficient CoT learning. Fig. 1 presents the key idea of our framework. Activation steering posits that LLM behaviors can be modulated via interventions along (approximately) linear directions in represen-

*Corresponding Author.

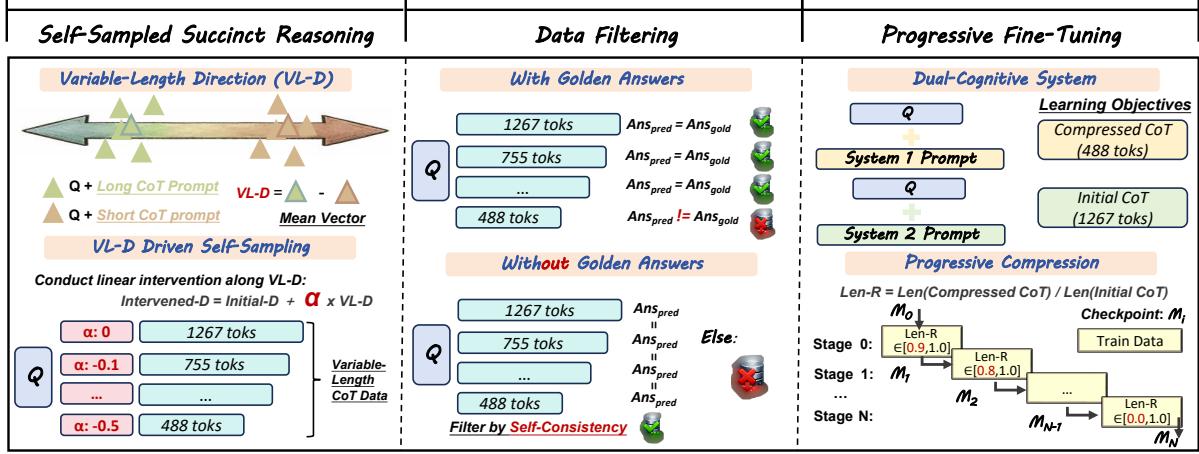


Figure 1: A self-sampling framework for efficient CoT learning. Our study (1) samples variable-length CoT data via intervention along VL-D; (2) filters data via answer or self-consistency verification; and (3) achieves efficient CoT internalization via a dual-cognitive system and progressive compression curriculum.

tation space (Tigges et al., 2023; Zou et al., 2023; Rimsky et al., 2024; Turner et al., 2023). Building on these insights, we conduct targeted analyses to identify a variable-length direction (VL-D) that governs CoT lengths. Guided by the intervention settings revealed in our probe analysis, we sample variable-length CoT traces directly from target LLMs themselves. To further ensure data quality, we apply either answer or self-consistency verification (Wang et al., 2022). Notably, for the latter, we retain prediction-consistent data of variable-length CoT variants, yielding a fully self-evolved data acquisition process. Our analysis shows that samples retained via self-consistency typically achieve near-perfect accuracy. During SFT, we adopt a dual-cognitive system and a progressively compressed curriculum, enabling LLMs to acquire fast-thinking capabilities while avoiding performance degradation caused by over-compression.

In our experiments, extensive evaluation on math and medical benchmarks shows that our method consistently outperforms prompt-control and SFT-based baselines and achieves performance competitive with RL-based baselines. Notably, our method exhibits strong adaptability across various LLMs (general and R1-style LLMs¹) and datasets, a setting that has rarely been validated in prior work. Overall, our contributions can be summarized as follows: 1) We propose S³-CoT to alleviate the data-scarcity bottleneck of SFT-based methods, via a standardized pipeline that samples high-quality, variable-length CoTs from target LLMs themselves.

¹In our study, we term LLMs that emit “<think></think>” reasoning traces as R1-style LLMs, and LLMs with standard outputs as general LLMs.

2) Leveraging self-sampled data, we enable efficient CoT internalization through SFT, providing an early exploration of LLM self-evolution. 3) Extensive experiments show our method achieves superior or competitive performance, with strong adaptability across various LLMs and datasets.

2 Related Work

2.1 Efficient CoT Internalization

Existing efforts largely fall into three paradigms: prompt control, SFT-based, and RL-based optimization. Prompt control imposes inference-time constraints by injecting explicit length cues or enforcing structured reasoning formats, offering a lightweight way to shorten CoT (Nayab et al., 2024; Renze and Guven, 2024; Xu et al., 2025; Han et al., 2025). SFT-based methods aim to internalize concise reasoning by fine-tuning LLMs on succinct CoT data, enabling shorter CoT without relying on prompts (Liu et al., 2024; Yu et al., 2024; Kang et al., 2025; Ma et al., 2025; Munkhbat et al., 2025; Xia et al., 2025). RL-based methods further treat conciseness as an optimization objective, explicitly balancing accuracy and length through reward design, and have shown strong effectiveness (Hou et al., 2025; Liu et al., 2025; Tu et al., 2025; Yi et al., 2025; Cheng et al., 2025; Luo et al., 2025a; Arora and Zanette, 2025). Such RL pipelines are most commonly applied to R1-style LLMs in existing work, whereas their applicability and stability for general LLMs are less explored. A more detailed description of existing methods can be found in our Appendix A.

2.2 Activation Steering

Activation steering (Zou et al., 2023) aims to control LLM behavior via intervention along approximately linear directions in representation space: Activation Addition demonstrates that adding a direction vector can induce target attributes (Turner et al., 2023), and Contrastive Activation Addition constructs steering vectors from contrastive residual differences to modulate behaviors like sycophancy (Rimsky et al., 2024). Related work further develops concept-direction discovery (e.g., sentiment directions (Tigges et al., 2023) and refusal directions (Arditi et al., 2024)), and inference-time interventions that change LLM outputs by targeted activation edits (Li et al., 2023; Azizi et al., 2025; Tang et al., 2025; Du et al., 2025).

3 Self-Sampled Succinct Reasoning

In this section, we primarily answer three questions: **Q1:** Does a length-controlled linear direction exist in LLMs’ representation space? **Q2:** How can we sample our expected data via intervention along this direction? **Q3:** Can this method facilitate the sampling of high-quality, variable-length CoT data? Our experiments are conducted primarily on GSM8K (Cobbe et al., 2021) and span both general LLMs (Qwen2.5_{7B} (Team, 2024) /LLaMA3_{8B} (Dubey et al., 2024)) and R1-style LLMs (DeepSeek-R1_{7B} (DeepSeek-AI, 2025)/Qwen3-Think_{4B} (Team, 2025)).

3.1 Identification of VL-D

VL-D Extraction. Prior direction-extraction methods typically derive representations from contrastive instruction pairs about a single attribute (Zou et al., 2023; Arditi et al., 2024; Tigges et al., 2023). Similarly, for length attribution, we append long and short CoT prompts to each instruction ($x \in D$), resulting in two sets D_L and D_S ($(x_l, x_s) \in (D_L, D_S)$). To formalize the direction extraction process, we begin with the decoder-only transformer architecture. Each input sequence $x = (x_1, x_2, \dots, x_n) \in \mathcal{V}^n$ is mapped to output probabilities ($y \in \mathbb{R}^{n \times |\mathcal{V}|}$). The residual stream activation of token i at the start of layer l is denoted as $\mathbf{h}_i^{(l)} \in \mathbb{R}^{d_{\text{model}}}$, initialized with its embedding $\mathbf{h}_i^{(1)} = \text{Embed}(x_i)$. Each layer applies both attention and MLP transformations: $\tilde{\mathbf{h}}_i^{(l)} = \mathbf{h}_i^{(l)} + \text{Attn}^{(l)}(\mathbf{h}_{1:n}^{(l)})$, $\mathbf{h}_i^{(l+1)} = \tilde{\mathbf{h}}_i^{(l)} + \text{MLP}^{(l)}(\tilde{\mathbf{h}}_i^{(l)})$. The variable-length direction can be extracted using the difference-in-means method (Marks and Tegmark, 2023; Panickssery

et al., 2023). For each layer $l \in [L]$ and final token position n , activations $(\mathbf{h}_n^{(l)}(x_l), \mathbf{h}_n^{(l)}(x_s))$ over $(x_l, x_s) \in (D_L, D_S)$ are obtained, and the corresponding difference-in-means vector can be calculated as: $\mathbf{u}^{(l)} = \mathbf{h}_n^{(l)}(x_l) - \mathbf{h}_n^{(l)}(x_s)$, $d^{(l)} = \mathbb{E}_{u \sim U^{(l)}}[u]$.

Visualization Analysis. To further investigate the nature of VL-D, we apply PCA (Hotelling, 1933) for dimensionality reduction and plot the extracted direction among each pair (x_l, x_s) . Here, we sample 100 data points from GSM8K, retaining only those where our appended CoT prompt significantly influences the length. Such a limitation allows us to filter out the impact of those cases where LLMs fail to follow instructions. As shown in Fig. 2, we present visualizations for the 6th and 15th layer under two LLMs, with visualizations for other LLMs and all layers provided in Fig. 4, Fig. 7, 8, 9, and 10. Two key phenomena emerge:

- **Layer-wise Separation Emergence:** Starting from the middle layers, a clear separation between x_l and x_s can be observed, while such separation is less pronounced in earlier layers. This suggests that the length-controlled directions may begin to emerge in the middle layers.
- **Parallelism of Directions:** Once the separation emerges, the directions extracted for each sample pair are highly parallel. This indicates the presence of a length-controlled direction, independent of the individual sample.

3.2 Intervention along VL-D

Quantitative Metrics. Although visualization analyses suggest the existence of variable-length directions, fine-grained intervention requires quantitative metrics to better understand their properties. Therefore, we introduce mean separation strength and angle variance metrics to monitor PCA-reduced features. The mean separation strength metric computes the L2 distance in each pair, and for the l^{th} layer, it can be calculated as:

$$\mathbf{S}^{(l)} = \frac{1}{|D_L|} \sum_{(x_l, x_s) \in (D_L, D_S)} \|\mathbf{h}_{pca}^{(l)}(x_l) - \mathbf{h}_{pca}^{(l)}(x_s)\|_2. \quad (1)$$

Meanwhile, the angle variance metric calculates the angle variance of each sample pair’s direction relative to their mean direction. For the l^{th} layer, by normalization, we can obtain unit vectors $\bar{\mathbf{u}}_i^{pca}$ for each pair and their mean \mathbf{v}^{pca} . The cosine value between each unit vector and the mean is: $\cos \theta_i = \bar{\mathbf{u}}_i^{pca} \cdot \mathbf{v}^{pca}$. The angle θ_i can be calculated

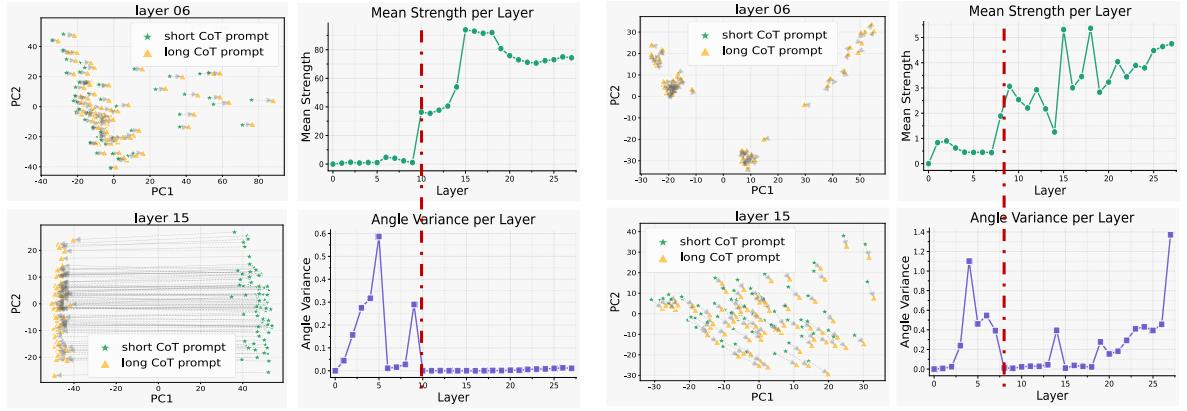


Figure 2: Analysis of VL-D properties. We provide PCA-based visualizations and quantify how the mean separation strength and angle variance metric vary across layers. Visualizations across all layers under various LLMs are in Fig. 7, 8, 9, and 10, respectively. Analysis on LLaMA3_{8B} and Qwen3-Think_{4B} are in Fig. 4.

by the inverse cosine function: $\theta_i = \arccos(\cos \theta_i)$ and the angle variance can be further calculated as:

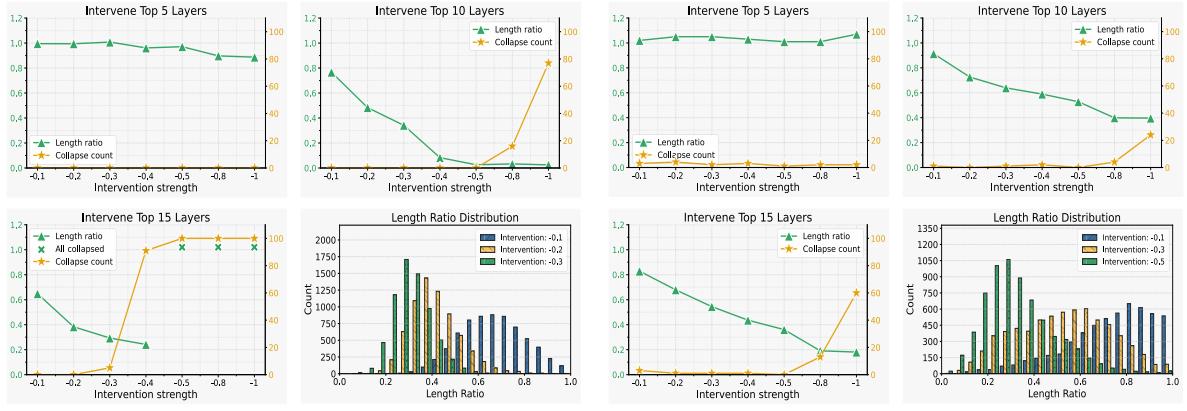
$$\sigma_\theta^2 = \frac{1}{m-1} \sum_{i=1}^m (\theta_i - \bar{\theta})^2. \quad (2)$$

where $\bar{\theta}$ represents the average value of θ . The line chart of Fig. 2 and Fig. 4 (in Appendix) presents the change of mean separation strength and angle variance across all layers. We observe that starting from certain middle layers (marked by the red dashed lines), there is a significant separation in each pair, and the directions among pairs become highly parallel. For Qwen2.5_{7B}, LLaMA3_{8B}, and Qwen3-Think_{4B}, this phenomenon persists until the final layer, while for Deepseek-R1_{7B}, it remains relatively stable in the middle layers. We guess that the extensive incremental training of DeepSeek-R1_{7B} may have introduced instability in its internal properties. Nevertheless, across various LLMs, we observe that there exists a length-controlled linear direction starting from the middle layers.

Probing Analysis. Our goal is to induce variable-length CoT data via interventions along VL-D. Following prior work (Arditi et al., 2024), the intervention can be modeled as a linear operation. Given an input x , we modify the hidden state at i^{th} token in l^{th} layer as: $\mathbf{h}_i^{(l)}(x) \leftarrow \mathbf{h}_i^{(l)}(x) + \alpha \times \mathbf{d}^{(l)}$, where α is a tunable intervention strength. Furthermore, we conduct a probing analysis to guide the choice of (i) intervention layers and (ii) intervention strength. Specifically, we define Length-Ratio (Len-R) as the ratio between the post-intervention output length and the initial length, which reflects the effectiveness of interventions. And we define

the layer at which the VL-D first emerges (marked by red dashed lines in Fig. 2 and 4) as the anchor layer. For layer selection, we intervene on a contiguous block of layers starting from the anchor layer. Taking Qwen2.5_{7B} as an example, the anchor layer is the 10th layer, “Top 1” corresponds to intervening on layers [10,11], and “Top 5” corresponds to [10, 15) layers, and so forth. For intervention strength, since our goal is to obtain shorter CoTs for efficient learning, we consider $\alpha \in \{-0.1, -0.2, -0.3, -0.4, -0.5, -0.8, -1\}$.

Our analysis evaluates an additional 100 samples, and Fig. 3 and Fig. 5 (in Appendix) summarize the probing results. We observe that: (1) weak interventions—either intervening on few layers or a small α —may fail to shorten CoTs, with Len-R remaining close to 1; (2) overly strong interventions—either intervening on many layers or a large α —may trigger generation collapse, resulting in repetitive outputs (marked by green “x”). These results highlight the importance of careful hyperparameter selection. Empirically, for general LLMs, intervening on top 5–10 layers with $|\alpha| \leq 0.5$ is typically stable, whereas for R1-style LLMs, intervening on top 15 layers with $|\alpha| \leq 0.5$ yields stable performance. While these trends provide coarse guidance, we do not observe a universally optimal setting across LLMs. Therefore, we advocate a standardized probing step on a small pilot set prior to large-scale intervention. According to probing results, to sample variable-length CoT data, our study intervenes on top 10/5/15/15 layers for Qwen2.5_{7B}/LLaMA3_{8B}/DeepSeek-R1_{7B}/Qwen3-Think_{4B} with $\alpha \in \{-0.1, -0.2, -0.3\}$,



(a) Analysis on Qwen2.5_{7B}.

(b) Analysis on Deepseek-R1_{7B}.

Figure 3: Probe experiments on intervention layers and strength. Green: average Len-R; Yellow: number of collapsed samples; Green “×”: all samples collapse. Bottom-right: Len-R distribution under large-scale sampling. Results for LLaMA3_{8B} and Qwen3-Think_{4B} are in Fig. 5, and results for other intervention settings are in Fig. 6.

({-0.1,-0.3,-0.5}), ({-0.1,-0.3,-0.5}), and ({-0.1,-0.3,-0.5}), respectively.

3.3 Verification of Data Quality

Since our sampled data are induced from target LLMs themselves, they can naturally keep style-aligned output. Accordingly, we focus on two other aspects of data quality: correctness and variable-length behavior.

For correctness. When gold answers are available, we adopt an answer verification scheme to filter data, retaining only those whose predictions match gold answers. However, in some practical settings, annotating answers is expensive. Therefore, as shown in Fig. 1, we explore a self-consistency verification scheme: only samples whose predictions remain consistent across variable-length variants are retained. Tab. 1 reports the number and the accuracy of GSM8K samples retained under this scheme. Interestingly, across various LLMs, retained samples typically achieve near-perfect accuracy. But the limitation of this scheme is that sampling efficiency will be affected by the underlying LLM capability. For example, for LLaMA3_{8B}, only 517 out of 6,838 samples are retained, whereas other stronger LLMs exhibit substantially higher sampling efficiency.

For variable-length behavior. The bottom-right corner of Fig. 3 and Fig. 5 (in Appendix) present the Len-R distribution of our sampled data. As the intervention strength $|\alpha|$ increases, the overall distribution shifts left, indicating shorter CoT on average. This trend confirms that our method can effectively sample variable-length CoT data.

| LLM | #Total | #Retained | #Correct | Acc. |
|---------------------------|--------|-----------|----------|--------|
| DeepSeek-R1 _{7B} | | 5,655 | 5,648 | 99.88% |
| Qwen3-Think _{4B} | 6,838 | 6,468 | 6,449 | 99.71% |
| Qwen2.5 _{7B} | | 4,564 | 4,560 | 99.91% |
| LLaMA3 _{8B} | | 517 | 516 | 99.81% |

Table 1: The number and accuracy of samples retained by self-consistency verification under various LLMs.

Overall, this section identifies the variable-length direction, conducts probe analysis on intervention settings, and validates the high quality of our sampled data.

4 Efficient CoT LLMs

In this section, we answer the question of whether self-sampled data can enable efficient CoT LLMs.

4.1 SFT Method

For SFT, our study adopts a dual-cognitive system and a progressive compression curriculum. Dual-cognitive theory suggests that human cognition comprises both fast thinking (System 1) and slow reasoning (System 2) (Evans, 2008). As illustrated in Fig. 1, we instantiate this framework by introducing the System 1 prompt and the System 2 prompt. Under the System 1 prompt, the learning objective is a compressed CoT, whereas under the System 2 prompt, the learning objective is the initial response. Meanwhile, our analysis (Sec. 5.2) shows that using the shortest CoTs to supervise SFT typically leads to over-compression, which significantly degrades LLM performance. To mitigate this issue, we adopt a progressive compression curriculum. As shown in Fig. 1, during training, we progressively expand the distribution of Len-R from ([0.9, 1.0]) to ([0.0, 1.0]) with a step size of

| Method | Accuracy↑ | | | | | Length↓ | | | | | AES↑ |
|-----------------------------------|-----------|--------|--------|--------|--------|---------|---------|---------|----------|---------|-------|
| | GSM8K | MATH | AMC23 | AIME24 | AVG. | GSM8K | MATH | AMC23 | AIME24 | AVG. | |
| Qwen2.5_{7B} | | | | | | | | | | | |
| Standard _p | 93.33% | 72.67% | 43.33% | 7.78% | 54.28% | 289.82 | 559.49 | 846.49 | 996.75 | 673.14 | - |
| Efficient _p | 89.67% | 70.00% | 44.17% | 7.78% | 52.90% | 107.15 | 300.83 | 573.44 | 741.21 | 430.66 | 0.11 |
| TokenSKIP | 90.83% | 74.67% | 45.00% | 2.33% | 53.21% | 260.63 | 512.12 | 766.95 | 842.71 | 595.60 | -0.08 |
| CoT-Valve | 90.50% | 71.00% | 37.50% | 6.68% | 51.42% | 298.82 | 619.19 | 900.23 | 1068.58 | 721.71 | -0.60 |
| C3oT | 93.50% | 71.33% | 51.67% | 5.56% | 55.52% | 291.22 | 536.77 | 788.60 | 866.80 | 620.85 | 0.19 |
| S ³ -CoT | 93.17% | 70.50% | 45.83% | 12.22% | 55.43% | 182.80 | 426.91 | 678.80 | 800.62 | 522.29 | 0.33 |
| S ³ -CoT ^{sc} | 92.50% | 69.67% | 46.67% | 11.11% | 54.99% | 184.10 | 433.43 | 687.33 | 831.47 | 534.08 | 0.27 |
| DeepSeek-R1_{7B} | | | | | | | | | | | |
| Standard _p | 93.33% | 92.33% | 90.83% | 51.11% | 81.90% | 1710.27 | 4261.18 | 6224.23 | 14061.22 | 6564.23 | - |
| Efficient _p | 84.33% | 90.33% | 83.33% | 47.78% | 76.45% | 511.83 | 3206.63 | 6062.81 | 11224.51 | 5251.44 | -0.47 |
| ShorterBetter | 79.33% | 72.33% | 66.67% | 37.78% | 64.03% | 140.59 | 585.57 | 1613.66 | 4701.88 | 1760.43 | -1.45 |
| LC-R1 | 85.43% | 88.67% | 85.00% | 42.22% | 75.33% | 449.44 | 1374.06 | 2788.19 | 6371.22 | 2745.73 | -0.22 |
| Eff _{Rea} | 91.67% | 91.33% | 88.33% | 53.33% | 81.17% | 1082.18 | 2700.98 | 4649.27 | 10706.94 | 4784.84 | 0.18 |
| LASER _{DE} | 93.00% | 91.33% | 88.33% | 51.11% | 80.94% | 974.57 | 1795.42 | 2766.08 | 5985.25 | 2880.33 | 0.44 |
| AutoTHINK | 92.83% | 93.67% | 88.33% | 47.78% | 80.65% | 1121.24 | 2449.16 | 3846.43 | 8303.59 | 3930.10 | 0.25 |
| CoT-Valve | 90.00% | 80.33% | 65.00% | 20.00% | 63.83% | 328.73 | 1499.40 | 1940.33 | 4177.39 | 1986.46 | -1.51 |
| C3oT | 92.50% | 92.00% | 87.50% | 51.11% | 80.78% | 1475.27 | 3805.73 | 6820.08 | 12884.91 | 6246.50 | -0.09 |
| S ³ -CoT | 91.17% | 92.00% | 90.83% | 51.11% | 81.28% | 1182.04 | 2833.27 | 5715.74 | 12217.53 | 5487.14 | 0.09 |
| S ³ -CoT ^{sc} | 91.67% | 90.67% | 87.50% | 50.00% | 79.96% | 1149.86 | 3016.64 | 5654.80 | 12167.95 | 5497.31 | -0.07 |

Table 2: Evaluation on math benchmarks under Qwen2.5_{7B} and DeepSeek-R1_{7B}.

0.1. At each iteration, we sample data to make Len-R as close to uniformly distributed as possible. Throughout compression, we follow a standard SFT pipeline, evaluate on a small validation set to select checkpoints.

4.2 Experiment settings

Training Data and LLMs. For training data, our study only uses the variable-length CoT data sampled from GSM8K as described in Sec. 3. We refer to an answer-verification based method as S³-CoT, and the self-consistency verification based method as S³-CoT^{sc}. We conduct experiments on both general LLMs (Qwen2.5_{7B}/LLaMA3_{8B}) and R1-style LLMs (Deepseek-R1_{7B}/Qwen3-Think_{4B}). On Qwen2.5_{7B} and Deepseek-R1_{7B}, we compare against state-of-the-art baselines, while on LLaMA3_{8B} and Qwen3-Think_{4B} we further demonstrate the adaptability of our method.

Baselines and Settings. For baselines, our study considers three families of methods: prompt control (Standard_p and Efficient_p (Renze and Guven, 2024)), SFT-based (TokenSkip (Xia et al., 2025), C3oT (Kang et al., 2025), and CoT-Valve (Ma et al., 2025)), and RL-based (ShorterBetter (Yi et al., 2025), LC-R1 (Cheng et al., 2025), Eff_{Rea} (Arora and Zanette, 2025), LASER_{DE} (Liu et al., 2025), and AutoTHINK (Tu et al., 2025)). Their implementation details are in Appendix B. Notably, RL-based methods are typically trained on DeepScaleR-Preview (Luo et al., 2025b), a composite dataset of 40K instances covering AIME,

AMC, MATH, etc. Their training process is highly compute-intensive, typically requiring 8×NVIDIA 80GB H100 GPUs. For our method, we adopt the LoRA framework (Hu et al., 2021) for SFT, and the LoRA hyperparameters are set to $r = 8$ and $\alpha = 16$. Our method just requires 2×NVIDIA 80GB A100 GPUs. For decoding, we follow the official generation settings, and the maximum generation length is set to 65,536.

Evaluation Data and Metrics. For evaluation data, we follow mainstream evaluation on math benchmarks, including GSM8K, MATH (Lightman et al., 2023), AMC23 (amc, 2023), and AIME24 (aim, 2024). Considering that RL-based methods are trained on a math mixture that may overlap with test distributions, we further assess generalization on cross-domain medical benchmarks, including MedQA (Jin et al., 2021), MedMCQA (Pal et al., 2022), and BULLET (Chen et al., 2025). A detailed description of evaluation data can be found in Appendix C. For the evaluation metric, we report accuracy averaged over three random responses, along with the corresponding average response token length. Moreover, we adopt the AES metric proposed in prior work (Luo et al., 2025a) to quantify the length-accuracy trade-off, computed as:

$$\text{AES} = \begin{cases} \omega \cdot \Delta\text{Length} + \beta \cdot |\Delta\text{Acc}|, & \text{if } \Delta\text{Acc} \geq 0, \\ \omega \cdot \Delta\text{Length} - \gamma \cdot |\Delta\text{Acc}|, & \text{if } \Delta\text{Acc} < 0. \end{cases} \quad (3)$$

where ΔLength represents the difference from the response length under Standard_p, ΔAcc represents

| Method | Accuracy↑ | | | | Length↓ | | | | AES↑ |
|-----------------------------------|-----------|---------|--------|--------|---------|---------|---------|---------|-------|
| | MedQA | MedMCQA | BULLET | Avg. | MedQA | MedMCQA | BULLET | Avg. | |
| Qwen2.5_{7B} | | | | | | | | | |
| Standard _p | 40.83% | 56.00% | 26.83% | 41.22% | 498.30 | 345.58 | 514.86 | 452.91 | – |
| Efficient _p | 49.17% | 60.33% | 31.67% | 47.06% | 118.43 | 62.23 | 107.11 | 95.92 | 1.50 |
| TokenSKIP | 45.67% | 51.33% | 23.17% | 40.06% | 461.66 | 335.16 | 464.99 | 420.60 | -0.21 |
| CoT-Valve | 55.00% | 61.00% | 40.33% | 52.11% | 564.33 | 411.82 | 584.32 | 520.16 | 1.17 |
| C3oT | 55.33% | 61.50% | 38.33% | 51.72% | 430.10 | 297.32 | 429.26 | 385.56 | 1.42 |
| S ³ -CoT | 54.50% | 60.17% | 34.83% | 49.83% | 302.19 | 192.57 | 309.47 | 268.07 | 1.45 |
| S ³ -CoT ^{sc} | 52.67% | 59.33% | 33.83% | 48.61% | 304.07 | 195.01 | 306.22 | 268.43 | 1.30 |
| DeepSeek-R1_{7B} | | | | | | | | | |
| Standard _p | 38.67% | 37.33% | 30.83% | 35.61% | 1865.66 | 1339.95 | 2169.36 | 1791.66 | – |
| Efficient _p | 36.83% | 34.33% | 27.50% | 32.89% | 1362.70 | 746.79 | 1297.02 | 1135.50 | -0.40 |
| ShorterBetter | 37.17% | 36.00% | 30.47% | 34.54% | 692.33 | 357.91 | 716.78 | 589.00 | 0.37 |
| LC-R1 | 35.83% | 37.50% | 29.17% | 34.17% | 1310.58 | 672.80 | 1248.25 | 1077.21 | -0.01 |
| Eff _{Rea} | 37.00% | 36.33% | 31.83% | 35.06% | 1736.26 | 948.04 | 1651.76 | 1445.35 | 0.04 |
| LASER _{DE} | 38.33% | 37.67% | 29.33% | 35.11% | 1354.27 | 868.28 | 1286.06 | 1163.54 | 0.21 |
| AutoTHINK | 39.50% | 38.33% | 31.33% | 36.39% | 1742.88 | 1117.90 | 1692.67 | 1517.82 | 0.26 |
| CoT-Valve | 28.00% | 30.83% | 24.67% | 27.83% | 1153.82 | 1676.99 | 918.82 | 1249.88 | -1.88 |
| C3oT | 36.50% | 36.83% | 32.67% | 35.33% | 1832.41 | 1265.82 | 1802.02 | 1633.42 | 0.01 |
| S ³ -CoT | 39.50% | 36.67% | 30.17% | 35.45% | 1648.16 | 1024.41 | 1608.56 | 1427.04 | 0.16 |
| S ³ -CoT ^{sc} | 40.00% | 38.50% | 28.17% | 35.56% | 1647.53 | 960.44 | 1555.83 | 1387.93 | 0.21 |

Table 3: Evaluation on medical benchmarks under Qwen2.5_{7B} and DeepSeek-R1_{7B}.

the difference from the accuracy under Standard_p, and ω , β , and γ are set to 1, 5, and 10, respectively.

4.3 Main Results

Compare with strong baselines. Tab. 2 and 3 summarize results on math and medical benchmarks. We observe that prompt control (Efficient_p) can shorten CoT length but often causes significant accuracy drops, motivating SFT/RL-based methods to internalize efficient CoT.

On math benchmarks, our method improves the overall accuracy-length trade-off. For Qwen2.5_{7B}, compared to Standard_p, it reduces length by ~150 tokens (~20% of initial length) while slightly increasing accuracy. Compared to strong baselines, our method achieves near-best accuracy while attaining the best AES, indicating a more balanced trade-off. For DeepSeek-R1_{7B}, our method compresses by ~1,100 tokens (~17% of initial length) with a small accuracy loss. Compared with SFT-based baselines (blue-shaded region), it yields markedly better AES: CoT-Valve over-compresses and hurts accuracy, whereas C3oT preserves accuracy but offers limited compression. Against RL-based baselines (green-shaded region), the accuracy of our method outperforms ShorterBetter and LC-R1 and is competitive with Eff_{Rea}, LASER_{DE}, and AutoTHINK, though behind them in length compression. We guess that this gap stems from potential train-test distribution overlap in RL-based methods. To enable a fair comparison, we further

evaluate on medical benchmarks to assess generalization under a shifted distribution.

On medical benchmarks, for Qwen2.5_{7B}, our method can compress by ~180 tokens (~40% of initial length) while achieving near-best accuracy, and remains among the most balanced methods with a strong AES. For DeepSeek-R1_{7B}, our method can compress by ~300 tokens (~17% of initial length) while maintaining accuracy. The most competitive baselines remain Eff_{Rea}, LASER_{DE}, and AutoTHINK. Compared with them, our method is essentially tied in accuracy and achieves comparable length compression—unlike the previously observed disadvantage. This result supports our guess that the earlier compression gap may stem from distributional overlap between the training and test data. Moreover, compared with other strong baselines, our method still shows clear advantages.

In aggregate, our method substantially outperforms SFT-based baselines without requiring external guidance, and matches RL-based baselines with fewer training resources. While we compare against RL-based methods separately, our method has the potential to be integrated with RL, serving as a warm-start (pre-training) stage before RL. We leave a thorough exploration of this integration to future work.

Adaptability across various LLMs. We further evaluate our method on LLaMA3_{8B} and Qwen3-Think_{4B} to assess cross-model adaptability. Tab. 4 reports results on math benchmarks. For

| Method | Accuracy↑ | | | | | Length↓ | | | | | AES↑ |
|-----------------------------------|-----------|--------|---------|--------|--------|---------|---------|----------|----------|---------|-------|
| | GSM8K | MATH | AMC23 | AIME24 | AVG. | GSM8K | MATH | AMC23 | AIME24 | AVG. | |
| LLaMA3_{8B} | | | | | | | | | | | |
| Standard _p | 78.83% | 47.67% | 20.00% | 3.33% | 37.46% | 245.06 | 605.28 | 857.21 | 1314.42 | 755.49 | - |
| Efficient _p | 68.67% | 43.00% | 24.17% | 3.33% | 34.79% | 109.61 | 422.35 | 708.14 | 937.38 | 544.37 | -0.43 |
| S ³ -CoT | 80.17% | 50.33% | 25.00% | 4.44% | 39.99% | 179.42 | 445.30 | 734.82 | 898.33 | 564.47 | 0.59 |
| S ³ -CoT ^{sc} | 79.67% | 49.33% | 22.50% | 4.44% | 38.99% | 176.11 | 496.30 | 699.45 | 1036.81 | 602.17 | 0.41 |
| Qwen3-Think_{4B} | | | | | | | | | | | |
| Standard _p | 96.00% | 93.00% | 99.17% | 82.22% | 92.60% | 1507.94 | 5573.16 | 10956.53 | 21009.29 | 9761.73 | - |
| Efficient _p | 94.33% | 91.33% | 96.67% | 76.67% | 89.75% | 812.05 | 4150.17 | 9344.19 | 18074.15 | 8095.14 | -0.76 |
| S ³ -CoT | 94.83% | 92.33% | 100.00% | 76.67% | 90.96% | 1029.56 | 4102.99 | 9180.60 | 17284.10 | 7899.31 | -0.41 |
| S ³ -CoT ^{sc} | 95.00% | 92.00% | 98.33% | 76.67% | 90.50% | 1061.49 | 4249.95 | 9162.59 | 17308.79 | 7945.71 | -0.54 |

Table 4: Evaluation on math benchmarks under LLaMA3_{8B} and Qwen3-Think_{4B}.

| Dataset | #Total | #Retained | #Correct | Acc. |
|---------|--------|-----------|----------|---------|
| PRM12K | 2,000 | 1,427 | 1,395 | 97.76% |
| MedQA | | 409 | 409 | 100.00% |

Table 5: For PRM12K and MedQA datasets, the number and accuracy of samples retained by self-consistency verification under DeepSeek-R1_{7B}.

| Method | PRM12K | | | MedQA | | |
|-----------------------------------|--------|---------|-------|--------|---------|------|
| | Acc.↑ | Len.↓ | AES↑ | Acc.↑ | Len.↓ | AES↑ |
| Standard _p | 81.90% | 6564.23 | - | 35.61% | 1791.66 | - |
| S ³ -CoT | 80.67% | 5206.07 | 0.06 | 35.14% | 1278.56 | 0.15 |
| S ³ -CoT ^{sc} | 79.33% | 5570.84 | -0.16 | 35.37% | 1238.28 | 0.24 |

Table 6: DeepSeek-R1_{7B} trained on PRM12K and MedQA, evaluated on math and medical benchmarks, respectively. We report the average accuracy and length.

LLaMA3_{8B}, our method compresses by ~160 tokens (~21% of initial length) while improving accuracy by 1–2% points. For Qwen3-Think_{4B}, it compresses by ~1,800 tokens (~18% of initial length) with a small accuracy drop. Combined with earlier results, these findings suggest that for general LLMs, our method can not only compress CoT length but also improve overall accuracy. But for R1-style LLMs, compression still incurs a slight accuracy trade-off, an open challenge shared by existing methods.

Overall, our experiments comprehensively validate the effectiveness, generalization, and adaptability of our method. In particular, S³-CoT^{sc}, serving as a fully self-evolving variant, exhibits substantial potential.

5 Analysis and Ablation

To provide deeper insight into our method, we present case studies in Appendix D. Here, we focus on answering the following two questions:

| Method | DeepSeek-R1 _{7B} | | | Qwen2.5 _{7B} | | |
|-----------------------|---------------------------|---------|-------|-----------------------|--------|-------|
| | Acc.↑ | Len.↓ | AES↑ | Acc.↑ | Len.↓ | AES↑ |
| S ³ -CoT | 81.28% | 5487.14 | 0.09 | 55.43% | 522.29 | 0.33 |
| Short _{only} | 74.89% | 4495.83 | -0.54 | 50.97% | 437.34 | -0.57 |

Table 7: Comparison against training with the shortest CoTs. We report the average accuracy and length.

5.1 Can our method generalize across various training datasets?

To answer this question, we run additional experiments on DeepSeek-R1_{7B} with PRM12K (math) (Lightman et al., 2023) and MedQA (medical) (Jin et al., 2021) as training data. Based on the obtained variable-length direction and intervention settings as described in Sec. 3, we sample 2,000 variable-length CoT instances per dataset. Fig. 11 (in Appendix) shows the resulting Len–R distributions, confirming that our method can still produce variable-length traces across datasets. Tab. 5 reports the number and accuracy of samples retained by self-consistency verification. Consistent with our earlier findings, this mechanism can help ensure the correctness of the sampled data: the retained MedQA samples even achieve 100% accuracy. Tab. 6 presents performance with our method trained on different datasets, and we observe that our method consistently yields substantial CoT length compression with minimal accuracy loss.

5.2 Why not use the shortest CoT samples as supervision?

Some prior work (Munkhbat et al., 2025; Kang et al., 2025) advocates supervising LLMs with the shortest possible CoT. In contrast, we find that—even with self-sampled data—training exclusively on the shortest CoT still leads to over-compression. As shown in Tab. 7, supervision with the shortest CoT achieves greater token com-

pression but substantially degrades accuracy. This observation suggests that, since the accuracy-preserving compression limit is unknown a priori, a progressive compression curriculum is necessary.

6 Conclusion

In summary, our study proposes a self-sampling framework (S^3 -CoT) for efficient CoT learning. We establish an end-to-end pipeline that guides how to sample high-quality, variable-length CoT from LLMs themselves, and extensive experiments demonstrate that our sampled data can enable efficient CoT LLMs. This line of exploration suggests an LLM-level capacity for self-evolution, and to the best of our knowledge, we are among the earliest teams to investigate this pathway. In future work, we will better leverage sampled data to push beyond the length–performance Pareto frontier.

Limitations

Our study alleviates the supervision data bottleneck in efficient CoT learning. However, how to more fully exploit the acquired variable-length data to push beyond the Pareto frontier between accuracy and length remains an open question. In particular, for R1-style LLMs, both our method and existing methods still face the challenge of slight accuracy degradation. Moreover, SFT-based methods can naturally serve as a warm-start (pre-training) stage before RL. Whether such an integration can yield additional performance gains is an important direction for future research.

References

2023. [math-ai/amc23](#). Hugging Face Datasets. Accessed: 2026-01-02.
2024. [math-ai/aime24](#). Hugging Face Datasets. Accessed: 2026-01-02.
- Andy Ardit, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083.
- Daman Arora and Andrea Zanette. 2025. Training language models to reason efficiently. [arXiv preprint arXiv:2502.04463](#).
- Seyedarmin Azizi, Erfan Baghaei Potraghloo, and Massoud Pedram. 2025. Activation steering for chain-of-thought compression. [arXiv preprint arXiv:2507.04742](#).
- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczek, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 17682–17690.
- Hanjie Chen, Zhouxiang Fang, Yash Singla, and Mark Dredze. 2025. Benchmarking large language models on answering and explaining challenging medical questions. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3563–3599.
- Zhengxiang Cheng, Dongping Chen, Mingyang Fu, and Tianyi Zhou. 2025. Optimizing length compression in large reasoning models. [arXiv preprint arXiv:2506.14755](#).
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. [arXiv preprint arXiv:2110.14168](#).
- DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. Preprint, [arXiv:2501.12948](#).
- Yanrui Du, Fenglei Fan, Sendong Zhao, Jiawei Cao, Qika Lin, Kai He, Ting Liu, Bing Qin, and Mengling Feng. 2025. Anchoring refusal direction: Mitigating safety risks in tuning via projection constraint. [arXiv preprint arXiv:2509.06795](#).
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. [arXiv preprint arXiv:2407.21783](#).
- Jonathan St BT Evans. 2008. Dual-processing accounts of reasoning, judgment, and social cognition. *Annu. Rev. Psychol.*, 59(1):255–278.
- Daya Guo, Dejian Yang, Huawei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shiron Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. [arXiv preprint arXiv:2501.12948](#).
- Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2025. Token-budget-aware llm reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24842–24855.
- Harold Hotelling. 1933. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

- Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. 2025. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. [arXiv preprint arXiv:2504.01296](#).
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. [arXiv preprint arXiv:2106.09685](#).
- 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](#).
- Di Jin, Eileen Pan, Nassim Oufattolle, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.
- Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. 2025. C3ot: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24312–24320.
- Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*.
- Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Cheng Jiayang, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2024. Can language models learn to skip steps? *Advances in Neural Information Processing Systems*, 37:45359–45385.
- Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. 2025. Learn to reason efficiently with adaptive length-based reward shaping. [arXiv preprint arXiv:2505.15612](#).
- 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](#).
- Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, and 1 others. 2025b. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. [Notion Blog](#).
- Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025. Cot-valve: Length-compressible chain-of-thought tuning. [arXiv preprint arXiv:2502.09601](#).
- Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. [arXiv preprint arXiv:2310.06824](#).
- 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](#).
- Sania Nayab, Giulio Rossolini, Marco Simoni, Andrea Saracino, Giorgio Buttazzo, Nicolamaria Manes, and Fabrizio Giacomelli. 2024. Concise thoughts: Impact of output length on llm reasoning and cost. [arXiv preprint arXiv:2407.19825](#).
- Ankit Pal, Logesh Kumar Umapathi, and Malaikanan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR.
- Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2023. Steering llama 2 via contrastive activation addition. [arXiv preprint arXiv:2312.06681](#).
- Matthew Renze and Erhan Guven. 2024. The benefits of a concise chain of thought on problem-solving in large language models. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, pages 476–483. IEEE.
- Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522.
- Xinyu Tang, Xiaolei Wang, Zhihao Lv, Yingqian Min, Wayne Xin Zhao, Binbin Hu, Ziqi Liu, and Zhiqiang Zhang. 2025. Unlocking general long chain-of-thought reasoning capabilities of large language models via representation engineering. [arXiv preprint arXiv:2503.11314](#).
- Qwen Team. 2024. *Qwen2.5: A party of foundation models*.
- Qwen Team. 2025. *Qwen3 technical report*. Preprint, [arXiv:2505.09388](#).
- Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. 2023. Linear representations of sentiment in large language models. [arXiv preprint arXiv:2310.15154](#).
- Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. 2025. Learning when to think: Shaping adaptive reasoning

in r1-style models via multi-stage rl. [arXiv preprint arXiv:2505.10832](#).

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. [arXiv preprint arXiv:2203.11171](#).

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

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. 2025. When more is less: Understanding chain-of-thought length in llms. [arXiv preprint arXiv:2502.07266](#).

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. 2025. Tokenskip: Controllable chain-of-thought compression in llms. [arXiv preprint arXiv:2502.12067](#).

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

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

Jingyang Yi, Jiazheng Wang, and Sida Li. 2025. Shorterbetter: Guiding reasoning models to find optimal inference length for efficient reasoning. [arXiv preprint arXiv:2504.21370](#).

Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024. Distilling system 2 into system 1. [arXiv preprint arXiv:2407.06023](#).

Yu Zhao, Hufeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2024. Marco-o1: Towards open reasoning models for open-ended solutions. [arXiv preprint arXiv:2411.14405](#).

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. [arXiv preprint arXiv:2310.01405](#).

A Detailed Descriptions of Existing Methods

Existing methods largely fall into three paradigms. **Prompt-control** constrains reasoning at inference by injecting explicit length signals or structured formats: TALE curbs overlong chains by estimating the token budget (Han et al., 2025), Concise Thoughts leverages cues (such as “Be concise.”) to bias LLMs toward shorter outputs (Nayab et al., 2024), and Chain-of-Draft encourages minimal intermediate draft notes to retain problem structure with substantially reduced verbosity (Xu et al., 2025). **SFT-based methods** fine-tune LLMs with succinct CoT as supervision: C3oT obtains compressed traces with the help of GPT-4o and trains LLMs on them (Kang et al., 2025). CoT-Valve learns a length-controllable LoRA module on QwQ_{32B} and scales the module strength to yield variable-length CoTs, which are used to distill other LLMs (Ma et al., 2025). **RL-based methods** optimize the length-accuracy trade-off with explicit reward signals (Luo et al., 2025a): ThinkPrune progressively tightens a hard token budget, penalizing trajectories that exceed the limit (Hou et al., 2025). LASER applies length-based reward and difficulty-aware variants to discourage overthinking on easy instances (Liu et al., 2025). ShorterBetter uses the shortest sample among multiple generations as a self-supervised target (Yi et al., 2025). LC-R1 combines a global length reward with an additional compression reward to remove redundant thinking (Cheng et al., 2025).

B Implementation of Strong Baselines

For prompt control, Standard_p uses standard prompting to elicit reasoning (“Please reason step by step, and put your final answer within \boxed{ }.”). And Efficient_p further imposes a conciseness constraint such as “Be concise.”. For SFT-based methods, we include TokenSkip, C3oT, and CoT-Valve: TokenSkip has released weights² trained on Qwen2.5_{7B}. For C3oT, we follow their settings, which prompt GPT-4o to remove redundant content. And for CoT-Valve, we adopt their provided variable-length GSM8K³ data sampled from QwQ-32B. Based on the obtained data, we guide SFT to re-implement their work according

²huggingface.co/hemingx/TokenSkip-Qwen2.5-7B-Instruct-GSM8K

³huggingface.co/datasets/horseee/MixChain-Z-GSM8K

to their settings. Notably, since CoT-Valve is based on QwQ-32B, the sampled data are typically longer than the default responses of Qwen2.5_{7B}. Consequently, after training, Qwen2.5_{7B}’s CoT tends to become longer rather than shorter. This outcome highlights the limitation of CoT-Valve: it does not universally compress CoT across all backbone LLMs. For RL-based methods, we consider ShorterBetter⁴, LC-R1⁵, Eff_{Rea}⁶, LASER_{DE}⁷, and AutoTHINK⁸, which all release weights on DeepSeek-R1_{7B}. We reproduce their results by following the provided reasoning templates and decoding configurations.

C Description of Evaluation Data

The description of our used evaluation data can be summarized as:

- **GSM8K:** A grade-school math word problem benchmark designed to evaluate multi-step numerical reasoning and arithmetic skills.
- **MATH:** A challenging competition-level mathematics dataset covering algebra, geometry, number theory, and calculus with step-by-step solution requirements.
- **AMC23:** A benchmark derived from the AMC 2023 competition, consisting of multiple-choice problems that test advanced pre-college mathematical reasoning.
- **AIME24:** A dataset based on the AIME 2024 exam, featuring short-answer problems that require precise symbolic reasoning and complex problem solving.
- **MedQA:** A large-scale medical question answering dataset composed of USMLE-style multiple-choice questions assessing professional-level clinical knowledge.
- **MedMCQA:** A medical multiple-choice QA benchmark sourced from Indian medical entrance exams, covering a broad range of clinical and basic medical topics.
- **BULLET:** A recent medical reasoning benchmark focused on evaluating LLMs’ robustness and generalization in complex, evidence-intensive clinical decision scenarios.

⁴huggingface.co/JingyangYi/SB_DS7B_alpha_2/tree/main

⁵huggingface.co/zx10086/LCR1_7B

⁶huggingface.co/daman1209arora/alpha_0.1_DeepSeek-R1-Distill-Qwen-7B

⁷huggingface.co/hkust-nlp/Laser-DE-L4096-7B/tree/main

⁸huggingface.co/SONGJUNTU/Distill-R1-7B-AutoThink-Stage3

In our study, to control evaluation cost, we randomly subsample the test sets to form our final evaluation set. Specifically, we sample 200 and 100 instances from GSM8K and MATH, respectively, and use the full test sets for AMC23 and AIME24. For the medical benchmarks (MedQA, MedMCQA, and BULLET), we randomly sample 200 instances from each dataset.

D Case Study

As shown in Fig. 12, we present case studies on Qwen2.5_{7B} and DeepSeek-R1_{7B}. For Qwen2.5_{7B}, our method can remove redundant phrasing while preserving the core reasoning steps, yielding a more concise CoT without affecting correctness. For DeepSeek-R1_{7B}, our method can further compress overly reflective behaviors. For example, the base LLM performs eight rounds of reflection when answering the given question. However, these reflections largely repeat the same viewpoint and amount to repeated self-verification. In contrast, our method can retain LLMs’ reflective ability while making the reflection more efficient and purposeful.

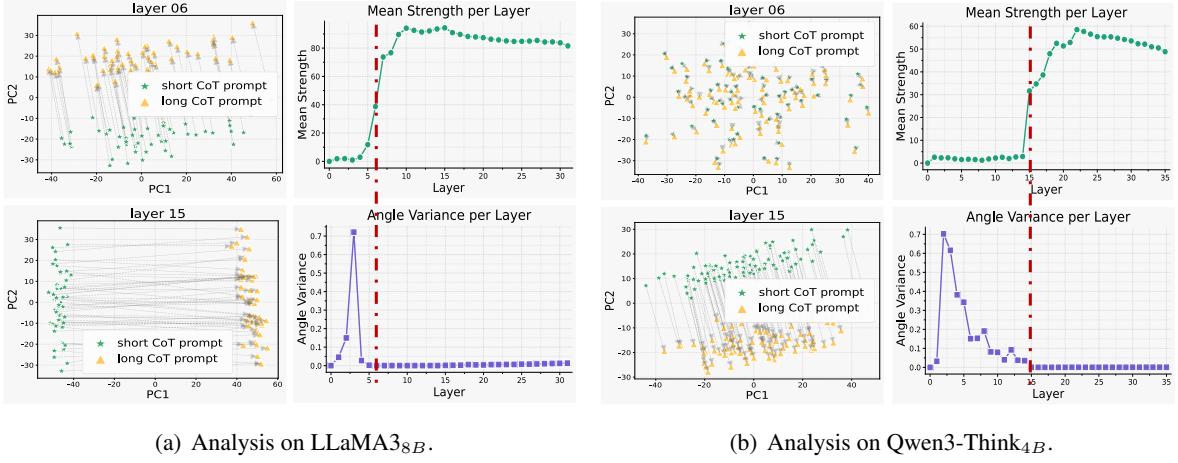


Figure 4: Analysis of VL-D properties under LLaMA3_{8B} and Qwen3-Think_{4B}. We provide PCA-based visualizations and quantify how the mean separation strength and angle variance metric vary across layers. Visualizations across all layers under various LLMs are in Fig. 7, 8, 9, and 10, respectively.

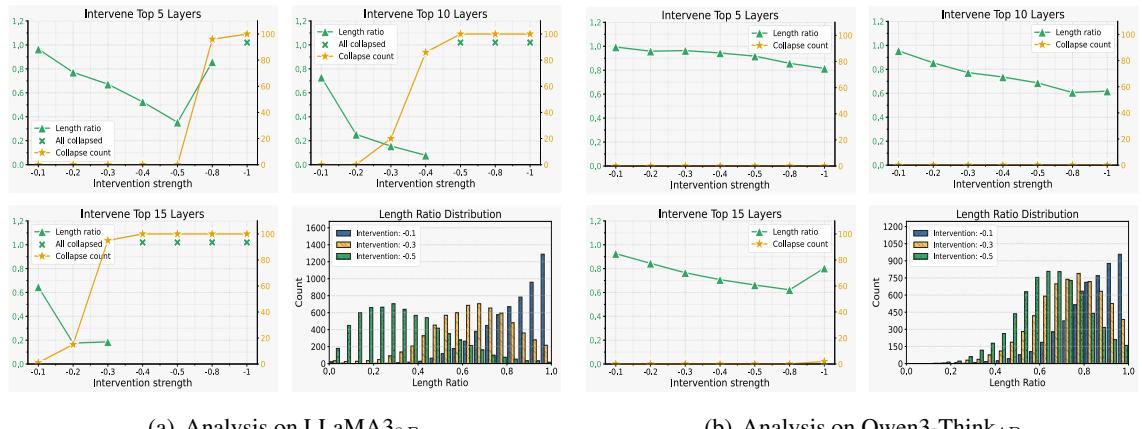


Figure 5: Probe experiments on intervention layers and strength under LLaMA3_{8B} and Qwen3-Think_{4B}. Green: average Len-R; Yellow: number of collapsed samples; Green “ \times ”: all samples collapse. Bottom-right: Len-R distribution under large-scale sampling. Results for other intervention settings are in Fig. 6.

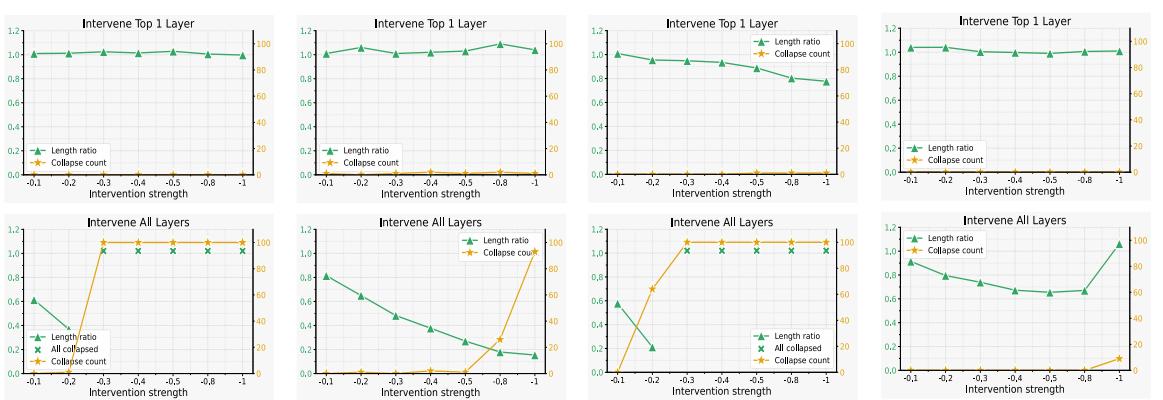


Figure 6: Results for different intervention settings under various LLMs. Green: average Len-R; Yellow: number of collapsed samples; Green “ \times ”: all samples collapse.

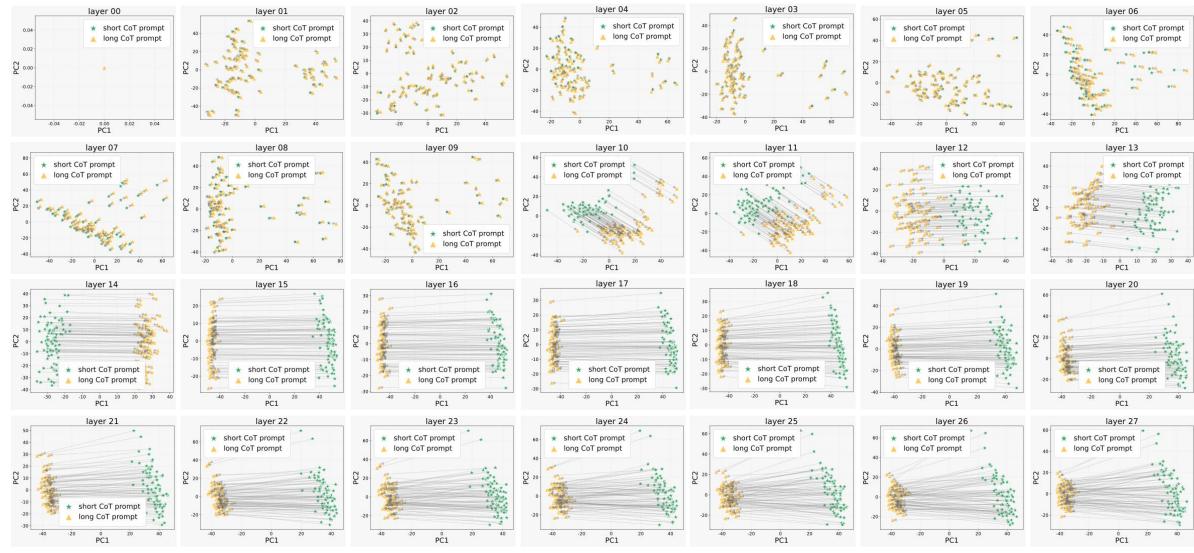


Figure 7: Visualizations across all layers under Qwen2.5_{7B}.

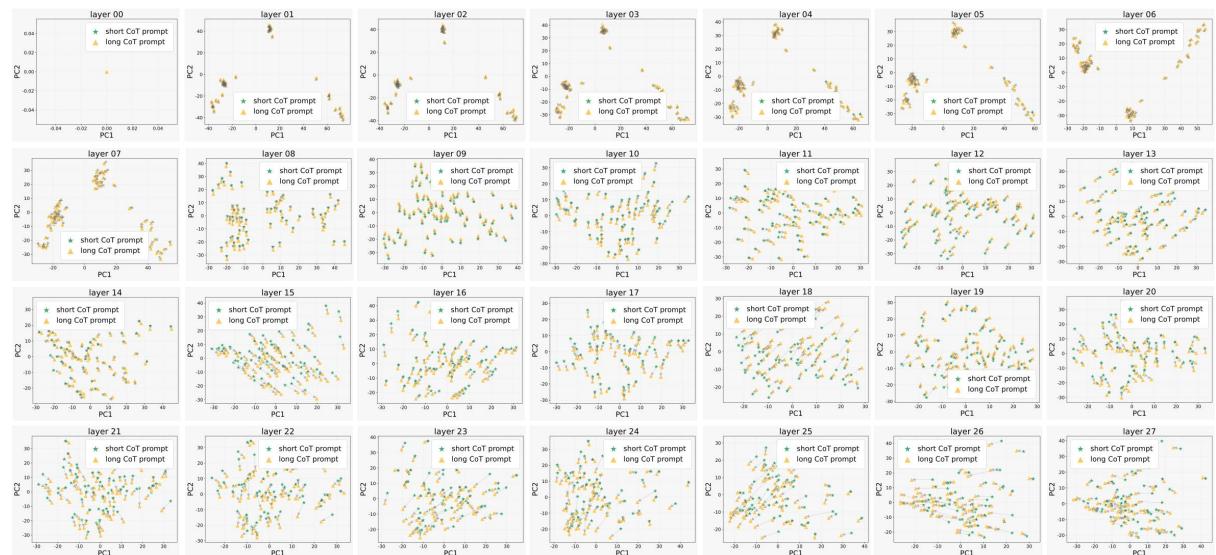


Figure 8: Visualizations across all layers under DeepSeek-R1_{7B}.

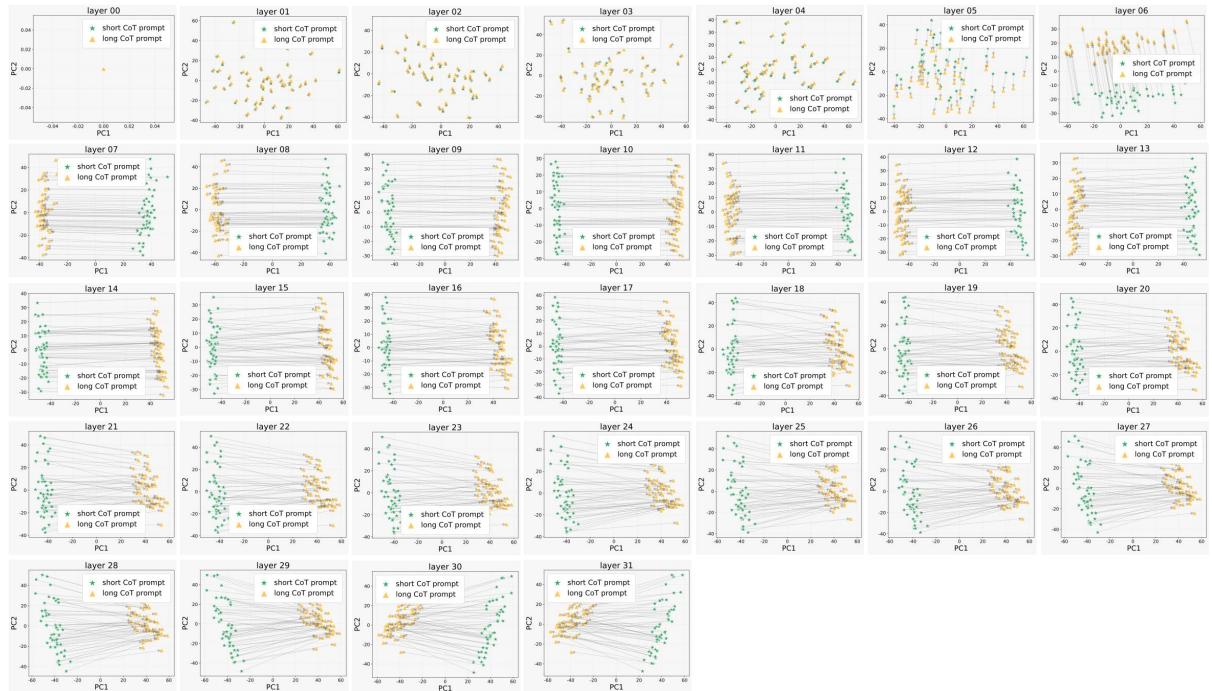


Figure 9: Visualizations across all layers under LLaMA3_{8B}.

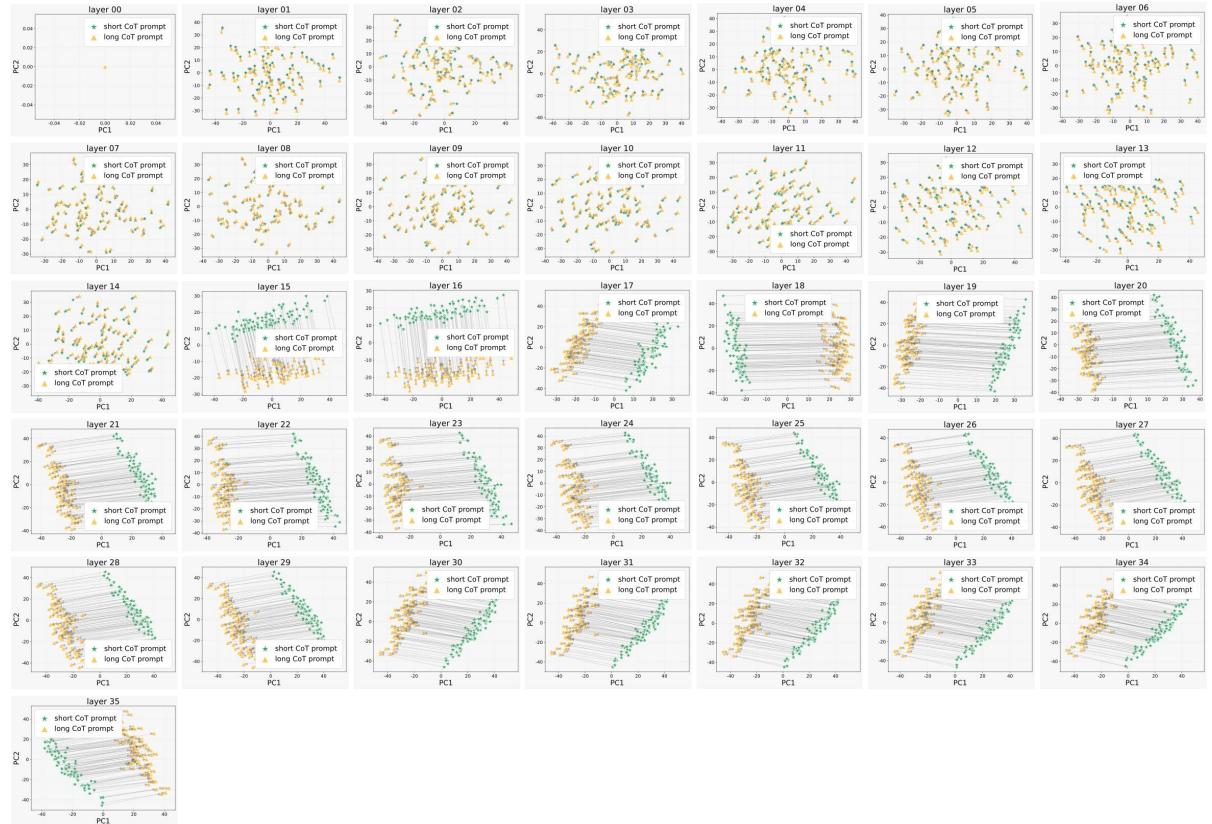


Figure 10: Visualizations across all layers under Qwen3-Think_{4B}.

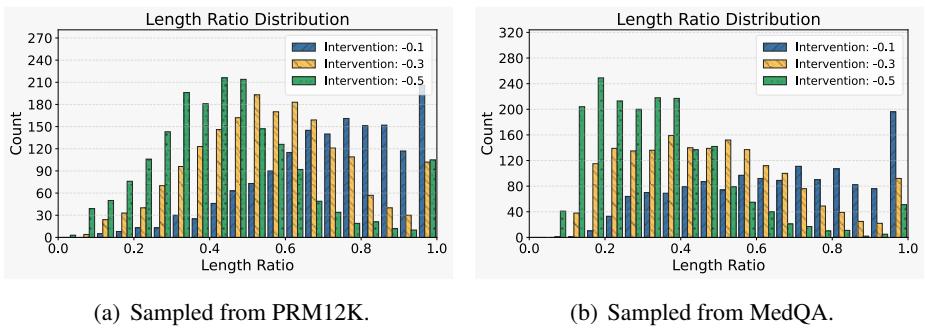


Figure 11: Len-R distributions of sampled data from PRM12K and MedQA, respectively. As the intervention strength increases, the overall distribution shifts left, indicating shorter CoT on average.

Question: While at the dollar store, Sloane counts 100 customers entering the store. The next day, she counts 50 more customers than the first day. If the total number of customers by the third day was 500, how many customers did she count on the third day?

Response (Base): Let's break down the problem step by step to find out how many customers Sloane counted on the third day.
1. **First Day:** Sloane counts 100 customers.
2. **Second Day:** She counts 50 more customers than the first day.
Therefore, the number of customers on the second day is $(100 + 50 = 150)$.
3. **Total Customers by the Third Day:** The total number of customers over the three days is 500.
Let's denote the number of customers on the third day as x .
The sum of the customers over the three days can be written as: $100 + 150 + x = 500$.
Solve for x : Combine the known quantities:
 $250 + x = 500$.
Subtract 250 from both sides to isolate x :
 $x = 500 - 250$.
Therefore, the number of customers Sloane counted on the third day is $\boxed{250}$.

Response (Our): Let's break down the problem step by step:
1. **First Day:** Sloane counts 100 customers.
2. **Second Day:** She counts 50 more customers than the first day. So, $(100 + 50 = 150)$ customers.
3. **Total by Third Day:** The total number of customers over three days is 500.
Let x be the number of customers on the third day. The total number of customers over three days is the sum of the customers from each day:
 $100 + 150 + x = 500$.
Solve for x :
 $250 + x = 500$.
 $x = 500 - 250$.
So, Sloane counted $\boxed{250}$ customers on the third day.

(a) Responses of Qwen2.5_{7B}.

Question: While at the dollar store, Sloane counts 100 customers entering the store. The next day, she counts 50 more customers than the first day. If the total number of customers by the third day was 500, how many customers did she count on the third day?

Response (Base): Okay, so I need to figure out how many customers Sloane counted on the third day. Let me try to break this down step by step.
First, the problem says that on the first day, she counts 100 customers. That seems straightforward. Then, the next day, which is the second day, she counts 50 more customers than the first day. Hmm, so if the first day was 100, then the second day would be 100 plus 50, right? Let me write that down:
First day: 100 customers
Second day: $100 + 50 = 150$ customers
Now, the problem mentions that by the third day, the total number of customers was 500. Wait, does that mean the total over three days is 500, or the number on the third day is 500? Hmm, I think it's the former because it says "the total number of customers by the third day was 500." So, that would be the sum of all customers from day one to day three.
So, let me define the variables:
day1 = 100
day2 = day1 + 50 = 150
day3 = ?
Total by day3 = day1 + day2 + day3 = 500
Now, if I plug in day1 and day2, I can solve for day3.
Calculating the total so far:
Total after two days = day1 + day2 = 100 + 150 = 250
Then, the total by the third day is 500, so the number of customers on the third day must be $500 - 250 = 250$.
Wait, is that right? Let me verify.
So, day1: 100, day2: 150, day3: 250. Adding those up: $100 + 150 = 250$, plus 250 is 500. Yes, that adds up correctly.
But hold on, let me make sure I'm interpreting the problem correctly. It says "the total number of customers by the third day was 500." That definitely means the sum of the first three days is 500. So, if the first two days sum to 250, then the third day must be $500 - 250 = 250$. So, that seems straightforward.
Alternatively, maybe I should think about whether the problem is asking for the number of customers on the third day or the total up to the third day. But the wording is "the total number of customers by the third day was 500," which is a bit ambiguous in terms of wording but I think in this context, it's referring to the cumulative total, not just the third day.
So, just to recap:
Day1: 100
Day2: 150
Total after Day2: 250
Total after Day3: 500
Therefore, Day3: $500 - 250 = 250$.
Hmm, that seems pretty clear. Is there another way to interpret it? Maybe the third day alone is 50 more than the second day? Let me check that.
Wait, the problem says: "The next day, she counts 50 more customers than the first day." So, the second day is 50 more than the first day. It doesn't say the third day is 50 more than the second day. So, the third day could be any number, but the total is given.
Therefore, if the total is 500, and the first two days add up to 250, then the third day must be 250.
To double-check, $100 + 150 + 250 = 500$.
Yep, that's correct.
I think that's the answer. So, Sloane counted 250 customers on the third day.
Final Answer: The number of customers Sloane counted on the third day is $\boxed{250}$.

Response (Our): Okay, so Sloane is counting customers at the dollar store over three days. On the first day, she counts 100 customers. The next day, she counts 50 more customers than the first day. So, that would be 100 plus 50, which is 150 customers on the second day.
Now, the total number of customers over the three days is 500. We already know the first two days: 100 and 150. Let me add those together: $100 + 150 = 250$. So, the first two days account for 250 customers.
To find out how many customers she counted on the third day, I can subtract the total of the first two days from the overall total. That would be 500 minus 250, which is 250.
Wait, so the third day also had 250 customers? That seems a bit low compared to the first two days, but maybe that's how it is. Let me double-check my calculations.
First day: 100 customers.
Second day: $100 + 50 = 150$.
Third day: Let's call it x . Total is $100 + 150 + x = 500$. So, $250 + x = 500$. Therefore, $x = 250$. Yeah, that seems correct.
So, Sloane counted 250 customers on the third day.
Final Answer: The number of customers Sloane counted on the third day is $\boxed{250}$.

(b) Responses of DeepSeek-R1_{7B}.

Figure 12: Case study on Qwen2.5_{7B} and DeepSeek-R1_{7B}. We highlight key reasoning steps and reflection steps in red.