

# How Do Answer Tokens Read Reasoning Traces? Self-Reading Patterns in Thinking LLMs for Quantitative Reasoning

Haoyang Chen<sup>†</sup> Yi Liu<sup>†</sup> Jianzhi Shao<sup>‡</sup> Tao Zhang<sup>§</sup> Chengfu Huo<sup>‡</sup> Wei Hu<sup>†,\*</sup>

<sup>†</sup> State Key Laboratory for Novel Software Technology, Nanjing University, China

<sup>‡</sup> Alibaba Group, China <sup>§</sup> Ant Group, China

chen7haoyang@gmail.com, yiliu07.nju@gmail.com, sjz250796@taobao.com

guyan.zt@antgroup.com, chengfu.huocf@taobao.com, whu@nju.edu.cn

## Abstract

Thinking LLMs produce reasoning traces before answering. Prior activation steering work mainly targets on shaping these traces. It remains less understood how answer tokens actually read and integrate the reasoning to produce reliable outcomes. Focusing on quantitative reasoning, we analyze the answer-to-reasoning attention and observe a benign self-reading pattern aligned with correctness, characterized by a forward drift of the reading focus along the reasoning trace and a persistent concentration on key semantic anchors, whereas incorrect solutions exhibit diffuse and irregular attention pattern. We interpret this as internal certainty during answer decoding, where the model commits to a viable solution branch and integrates key evidence. Following this, we propose a training-free steering method driven by Self-Reading Quality (SRQ) scores combining geometric metrics for process control with semantic metrics for content monitoring. SRQ selects data to build steering vectors that guide inference toward benign self-reading and away from uncertain and disorganized reading. Experiments show that our method yields consistent accuracy gains.

## 1 Introduction

Thinking LLMs (a.k.a. reasoning models), such as DeepSeek-R1, GPT-5, and Gemini 3 series, exhibit strong quantitative reasoning abilities and typically generate a reasoning trace (e.g., separated by `</think>`) before the final answer (Wu et al., 2025). To control these models, activation steering emerges as a powerful, training-free intervention method for modulating latent representations. While proven robust for general alignment tasks like eliciting honesty (Eshuijs et al., 2025) and ensuring instruction following (Stolfo et al., 2025), recent work adapts steering techniques to optimize

the reasoning traces of thinking LLMs for interpreting internal mechanisms (Venhoff et al., 2025), compressing verbose outputs (Azizi et al., 2025), eliciting extended capabilities (Zhao et al., 2025b), and calibrating reliability (Chen et al., 2025).

Despite these efforts, *it remains unclear how answer tokens actually read the reasoning trace*. An inspiring analysis (Zhang et al., 2025) confirms the existence of attention links between answer tokens and reasoning tokens, but there is still a lack of exploration of how answer tokens navigates noise and leverages key information from the reasoning trace during decoding. This issue is particularly important in thinking LLMs where reasoning chains often span thousands of tokens.

To understand how answer tokens read and utilize reasoning tokens, we investigate the self-reading behavior of thinking LLMs by conducting an attention analysis on quantitative reasoning tasks. We identify a stable, structured reading pattern emerging in the middle-to-late layers, correlating strongly with correctness. In correct samples, rather than scanning the entire trace indiscriminately, the answer tokens exhibit a *benign* self-reading mode characterized by two features: (i) a progressive forward shift of the attention centroid (or reading focus) toward later reasoning positions as decoding proceeds, and (ii) a persistent concentration on key semantic anchors, such as constraints, solution plans, and conclusions.

Figure 1 visualizes a typically benign self-reading pattern observed in a correct solution. In the answer panel, the shading shows that the reading focus of answer tokens moves forward along the reasoning trace: bluer tokens attend to earlier positions (e.g., “Angles A and B are supplementary”), while purpler tokens attend later steps (e.g., “\boxed{160} degrees”). The reasoning panel highlights intense attention on key semantic steps (e.g., “Recall knowledge” and “Plan the solution”). As the answer is decoded, the dependency centroid

\* Corresponding author

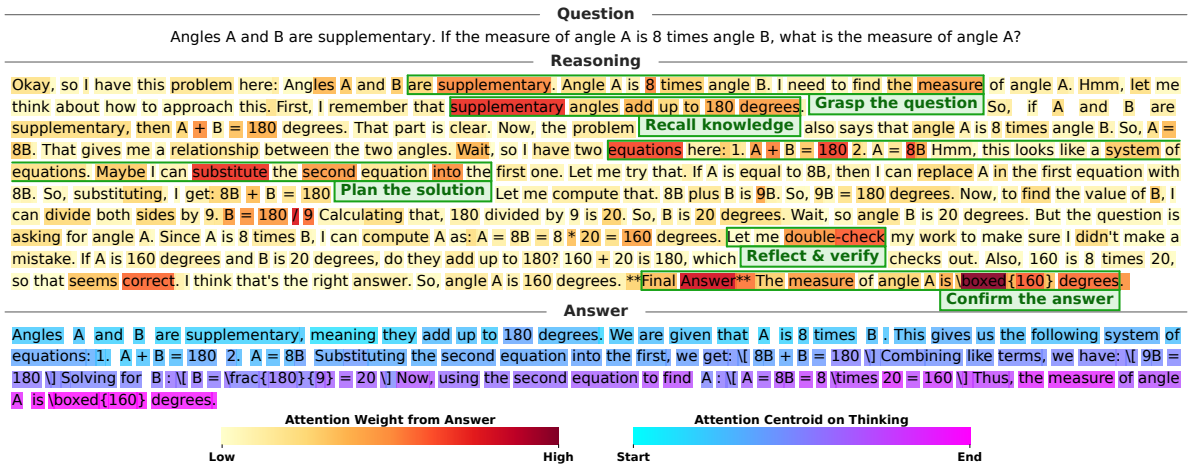


Figure 1: Visualization of benign self-reading in the middle-to-late layers of R1-Distill-Llama-8B. The example is selected from the GSM8K (Cobbe et al., 2021) benchmark. The answer panel colors tokens from blue to purple based on the attention centroid location to show the forward shift. The centroid is defined as the weighted average center of an answer token’s attention distribution across reasoning tokens. The reasoning panel uses a gradient from yellow to red to indicate stronger attention accumulation from the answer tokens on pivotal semantic anchors.

gradually shifts toward later reasoning positions. Simultaneously, attention repeatedly returns to key semantic anchors like reflections and conclusions.

These behaviors reveal that thinking LLMs strategically read their own reasoning traces during answer generation. We interpret this benign self-reading as a behavioral signature of internal certainty: by the answer stage, the model has committed to a particular solution path and relies on a few key reasoning steps, e.g., problem constraints and conclusions, as the evidence for answer generation. This interpretation aligns with classic cognitive theories (Nelson, 1990; Koriat, 1997) that cognition involves a meta-level that processes information derived from object-level execution based on internal states by *control* and *monitoring*. In this analogy, reasoning tokens implement object-level computation, while answer tokens function as a meta-level operation that reads and distills the reasoning trace to support generation. The progressive forward drift of the attention centroid reflects *control*: the model advances its reading focus along a chosen branch instead of wandering. Meanwhile, the persistent focus on semantic anchors implements *monitoring* by repeatedly revisiting the evidence that supports the answer tokens. Benign self-reading thus marks a stable, orderly, and internally confident state that typically coincides with correct solutions. In contrast, disorganized attention in incorrect solutions signals cognitive uncertainty. Section 5.7 will show an example.

Building upon these insights, we propose a steer-

ing method driven by Self-Reading Quality (SRQ) scores to formalize and promote benign reading behaviors. SRQ measures the quality of a model’s self-reading from both geometric and semantic dimensions. The geometric dimension captures the process structure of control strategies, where metrics represent decisive execution along a solution path. The semantic dimension assesses the content grounding of monitoring quality by verifying whether the attended reasoning token is a critical step. By selecting samples with different SRQ scores, we construct steering vectors that guide LLMs toward benign self-reading and away from disorganized patterns associated with cognitive uncertainty. Our experiments on a range of thinking LLMs show that our method consistently boosts accuracy by steering the models toward a more decisive and grounded answer stage.

Our main contributions are outlined as follows:

- We identify the benign self-reading pattern during answer decoding on quantitative reasoning tasks. The pattern reflects the internal certainty and affects correctness.
- We propose an SRQ-driven steering method. It guides the model toward more certain and well-organized self-reading patterns.
- We evaluate our method on multiple quantitative reasoning benchmarks. Results show an accuracy improvement up to 2.6% over the base LLMs, confirming the link between self-reading and answer correctness.

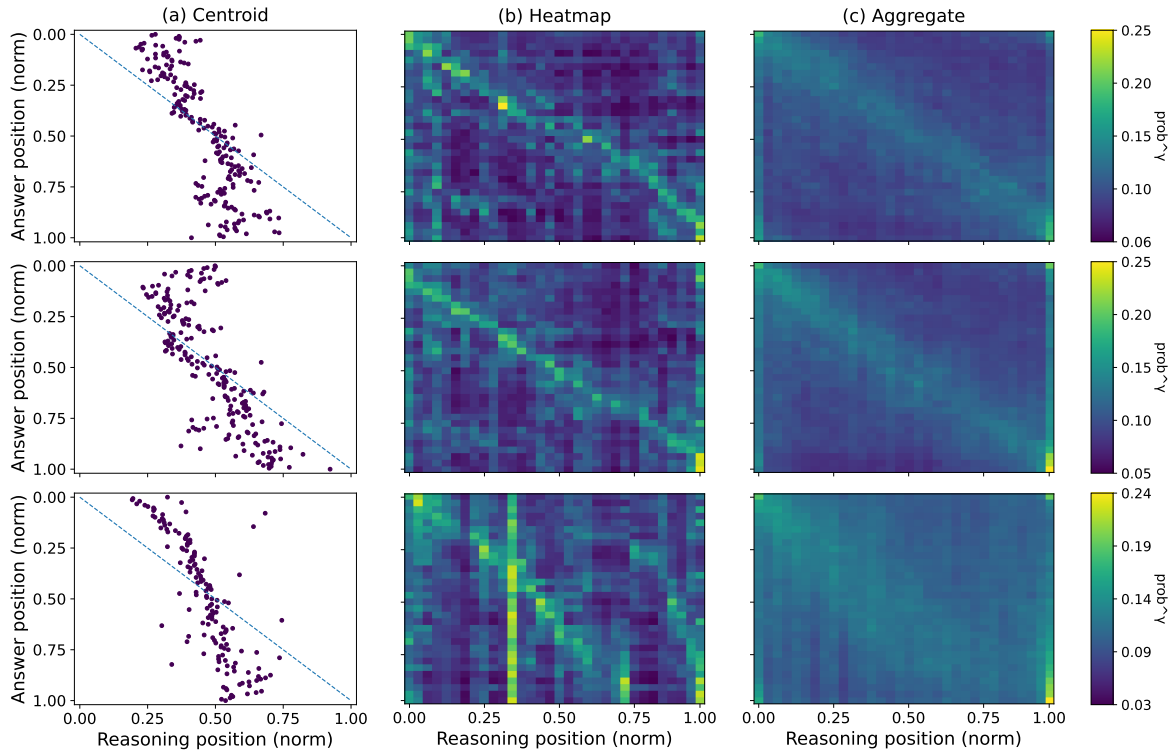


Figure 2: Attention visualization of benign self-reading on GSM8K. Rows show R1-Distill-Llama-8B, R1-Distill-Qwen-7B, and Qwen3-4B-Thinking. Column (a) displays attention centroids, where each point represents the weighted average center of an answer token’s attention distribution over the reasoning. Columns (a) and (b) use the same problem, while Column (c) shows the aggregated heatmaps averaged over 100 correct solutions per model.

## 2 Related Work

Activation engineering steers language model outputs by modifying intermediate activations during inference. Turner et al. (2023) propose activation addition, an inference-time method that builds a steering vector from contrastive samples and adds it to hidden states to control attributes such as sentiment and toxicity. Tang et al. (2025) identify reasoning critical feed forward neurons based on activation differences in strong and weak reasoning traces. Eshuijs et al. (2025) use steering vectors to generate safety steered honest alternatives, which helps LLM judges detect dishonesty reliably. Stolfo et al. (2025) compute directions from activation differences with and without instructions and steer models to satisfy constraints on format length and required words, even with insufficient prompts.

Recent studies investigate activation engineering within thinking LLMs that separate reasoning and answering stages. Venhoff et al. (2025) extract steering vectors from DeepSeek-R1-Distill models to steer example-based verification and backtracking. They find that these behaviors align with simple linear directions in activation space. Azizi

et al. (2025) propose steered compression, extracting a vector that moves generation from verbose to concise reasoning and injecting it to reduce reasoning length while largely preserving accuracy. Li et al. (2025) leverage the activation difference between symbolic mathematical generations and natural language generations to steer the model toward enhanced logical performance. Sun et al. (2025) identify a direction that controls reasoning length and edit projection weights of attention heads. As far as we know, these works have neither analyzed nor steered how answer tokens read the reasoning trace during answer decoding.

## 3 Self-Reading in Thinking LLMs

We conduct an analysis of the answer-to-reasoning attention mechanism on the GSM8K quantitative reasoning task (Cobbe et al., 2021). We utilize three frontier thinking LLMs: R1-Distill-Llama-8B (DeepSeek-AI, 2025), R1-Distill-Qwen-7B (DeepSeek-AI, 2025), and Qwen3-4B-Thinking (Yang et al., 2025a). We focus on the middle-to-late layers, which are regarded more semantically rich and stable (Langedijk et al., 2024; Skean et al., 2025; Zhang et al., 2025).

Type	$C_{SR+}$	$C_{SR-}$	$I_{SR+}$	$I_{SR-}$	$Dis$
Count	159	12	3	23	$2^+ + 1^-$

Table 1: Human annotations on 200 randomly sampled GSM8K solutions.

### 3.1 Main Analysis

We find that correct solutions exhibit a structured self-reading pattern during answer decoding, characterized by two distinct features:

**Forward-shifting attention centroid.** First, the reading focus of answer tokens progressively shifts from earlier to later reasoning positions as decoding proceeds. To quantify this, we normalize the attention scores of each answer token over the thinking tokens and compute the weighted average position (centroid) of its focus. As visualized in Figure 2(a), with positions normalized to  $[0, 1]$ , the resulting trajectory follows a clear diagonal path, indicating that the attention centroid moves synchronously with the generation of the answer. Figure 2(b) also shows that the reading focus of answer tokens advances along the reasoning positions, forming a diagonal trend across models, with occasional parallel secondary bands. The forward-shift of reading focus suggests that the model captures global information by following the logical flow of the reasoning trace. It indicates that the model deterministically tracks an effective solution path. We hypothesize that this sequential alignment implies that the reasoning content itself is logically coherent and the smooth synthesis of prior thoughts during the answer generation. See Appendix A.3 for the contrasting visualizations of failure cases.

**Semantic anchor concentration.** Beyond the global drift, the model relies on key semantic information in the reasoning trace to generate answer tokens. We observe that the answer tokens repeatedly concentrate on key *semantic anchors*, such as problem constraints, solution plans, reflections, and final conclusions, revisiting a small set of crucial steps. As shown in the heatmaps of Figure 2(b), this behavior manifests as high-attention hotspots along the diagonal as well as bright off-diagonal anchors. These local highlights underscore that the model strategically retrieves and reinforces key details rather than spreading attention uniformly. Together with the two features, the benign self-reading serves as a signature of the model’s *internal certainty* when generating the answer.

To verify the universality of the benign self-

Type	$C_{SR+}$	$C_{SR-}$	$I_{SR+}$	$I_{SR-}$	$Dis$
Count	48	2	4	46	0

Table 2: Human annotations on a balanced subset of GSM8K samples, including 50 correct and 50 incorrect solutions.

reading patterns, we aggregate the attention maps from 100 correctly solved samples. In Figure 2(c), the aggregated heatmaps maintain a distinct, soft diagonal ridge across all models. This confirms that the structured self-reading pattern is not an anecdotal but a stable and consistent behavior intrinsic to correct reasoning in thinking LLMs.

### 3.2 Benign Self-Reading and Correctness

To validate the link between self-reading and accuracy, we conduct a human assessment on 200 solutions from R1-Distill-Llama-8B. We assign an example to one of four categories only when all three human annotators agree: correct with benign self-reading ( $C_{SR+}$ ), correct without benign self-reading ( $C_{SR-}$ ), incorrect with benign self-reading ( $I_{SR+}$ ), and incorrect without benign self-reading ( $I_{SR-}$ ). Examples with disagreement are grouped into a separate category ( $Dis$ ).

We first annotate 200 randomly sampled solutions. Table 1 shows that benign self-reading is common in correct solutions but rare in incorrect ones. Since this random sampling is naturally imbalanced, we further annotate a balanced subset with equal numbers of correct and incorrect solutions. The result in Table 2 confirms the same trend, indicating that benign self-reading is closely associated with correct and internally stable reasoning. Appendix A gives more analysis.

## 4 Self-Reading Steering Method

Motivated by our analysis above, we propose a steering method driven by *Self-Reading Quality* (SRQ) scores, which quantify how effectively answer tokens read the model’s reasoning traces.

### 4.1 From Self-Reading to Internal Certainty

Early research (Nelson, 1990; Koriat, 1997) in cognitive psychology characterizes meta-cognition as an interaction between an *object level* that executes the primary task and a *meta level* that processes available information via *monitoring* and *control*. In thinking LLMs, reasoning tokens implement the object-level operation to reason, while answer

tokens that read and summarize the generated reasoning trace acting as the meta-level operation.

Benign self-reading in thinking LLMs aligns with this meta-cognitive theory through two features: (i) A forward-shifting reading focus acts as *control*: the meta level utilizes object-level information by steadily advancing attention along a chosen solution path instead of wandering around irrelevant regions. (ii) High-attention anchors on key reasoning steps serve as *monitoring*: the meta level inspects important steps of the reasoning to evaluate and support answer generation. This structured control and monitoring indicate a natural signature of *internal certainty*.

Accordingly, we decompose SRQ into a *geometric* dimension that measures process control (how consistently attention progresses along the reasoning trace) and a *semantic* dimension that measures content monitoring (whether attention concentrates on high-value semantic anchors and avoids misleading regions). We formalize the two dimensions below and show how the resulting SRQ scores enable reliable sample selection and effective steering.

## 4.2 Geometric Self-Reading Quality

The *geometric* dimension measures whether answer tokens commit to a solution path and move steadily forward along the reasoning trace instead of wandering. Specifically, we use the *global* and *local* metrics to evaluate the forward diagonal trend of attention centroid movement.

To quantitatively track the reading focus of answer tokens over the reasoning trace, we define the *attention centroid*. Let  $P \in \mathbb{R}^{A \times T}$  be the answer-to-reasoning attention submatrix, with  $A$  answer tokens and  $T$  reasoning tokens. We form a row-normalized version  $\tilde{P}_{ij} = \frac{P_{ij}}{\sum_{k=1}^T P_{ik} + \varepsilon}$ , so each answer token defines a distribution over reasoning positions. We index answer and reasoning positions by their normalized coordinates  $x_i = \frac{i}{A}$  and  $t_j = \frac{j}{T}$ . We define the attention centroid of answer token  $i$  on the reasoning trace as  $y_i = \sum_{j=1}^T t_j \tilde{P}_{ij}$ . The sequence  $(x_i, y_i)$  tracks how the reading focus moves during answer decoding.

**Global forward-reading pattern.** To capture the smooth forward drift of the centroid in benign self-reading, we define two global scores. The first is the *Pearson correlation* between  $x$  and  $y$ :

$$SRQ_{\text{corr}} = \text{corr}(x, y), \quad (1)$$

where  $x = (x_1, x_2, \dots, x_A)$  and  $y = (y_1, y_2, \dots,$

$y_A)$ . It achieves maximal when later answer tokens read later reasoning tokens.

The second measures how closely the centroid trajectory follows the diagonal of the attention map:

$$SRQ_{\text{diag}} = 1 - \sqrt{\frac{1}{A} \sum_{i=1}^A (y_i - x_i)^2}. \quad (2)$$

**Local forward-reading consistency.** While the global scores capture the diagonal trend, benign self-reading also exhibits locally forward-moving focus. We measure local consistency using a sliding-window of size  $w$  coverage score, which evaluates whether high-attention points exhibit a forward progression. For each answer token, we extract its high-attention points by selecting the minimal set of reasoning positions with the largest  $\tilde{P}_{ij}$  whose cumulative sum reaches  $\beta$ . Each point maps to the coordinates of normalized answer and reasoning positions. Then, for each of the  $U$  overlapping windows, where  $U = A - w + 1$ , we compute a *Pearson correlation*  $r_k$  between these coordinates.  $SRQ_{\text{local\_cover}}$ , representing the fraction of windows exceeding a threshold  $\tau$ , is defined as:

$$SRQ_{\text{local\_cover}} = \frac{1}{U} \sum_{u=1}^U \mathbb{1}(r_k > \tau). \quad (3)$$

## 4.3 Semantic Self-Reading Quality

On the *semantic* dimension, answer attention is expected to repeatedly return to key semantic anchors rather than on noisy or incorrect ones. For supervision, we annotate each solution with span-level labels by LLM APIs (e.g., GPT-5 and Gemini-3-pro-preview) and project them to token indices. This yields a set  $G$  of good reasoning tokens (e.g., constraints, correct intermediate steps), a set  $B$  of bad reasoning tokens (e.g., incorrect computations, misleading branches), and a set  $K_{\text{ans}}$  of key answer tokens such as the final numerical results.

**Semantic quality of reasoning focus.** We first measure how the answer stage distributes its total attention flow over good and bad reasoning steps. Based on the raw matrix  $P$ , we compute a column sum  $c_j = \sum_{i=1}^A P_{ij}$  for each reasoning token  $j$  and select the top  $\gamma T$  columns with the largest  $c_j$  as a high-flow set  $H$ . The fraction on good ( $G$ ) or bad ( $B$ ) tokens is

$$SRQ_{\text{think}}^{\pm} = \frac{\sum_{j \in H \cap S} c_j}{\sum_{j \in H} c_j}, \quad S \in \{G, B\}. \quad (4)$$

**Support quality of the answer.** We measure whether key answer tokens attend to correct supporting steps. Given  $K_{\text{ans}}$ , we define  $SRQ_{\text{ans}}^{\pm}$  as the average attention that they allocate to good or bad tokens:

$$SRQ_{\text{ans}}^{\pm} = \frac{\sum_{i \in K_{\text{ans}}} \sum_{j \in \mathcal{S}} \tilde{P}_{ij}}{|K_{\text{ans}}|}, \quad S \in \{G, B\}. \quad (5)$$

**Boundary emphasis on constraints and conclusions.** Benign self-reading also concentrates attention on problem constraints and final conclusions. Using  $P$ , we define  $m_{\text{start}}$  and  $m_{\text{end}}$  as the attention sum on the first and last  $\rho_{\text{bd}}$  fraction of reasoning tokens. Instead of a uniform allocation giving  $\rho_{\text{bd}}$ , we target mild enrichment with  $\rho_{\text{tar}} = \alpha \rho_{\text{bd}}$  ( $\alpha > 1$ ). We score deviation from this target as follows:

$$SRQ_{\text{start/end}} = \frac{m_{\text{start/end}} - \rho_{\text{tar}}}{1 - \rho_{\text{tar}}}. \quad (6)$$

#### 4.4 SRQ-Based Sample Selection and Steering

With the geometric and semantic metrics, we build an integrated SRQ score per solution, and use it to select the contrastive sets and steering directions.

**From metrics to SRQ.** For each solution  $n$ , we first rescale each geometric or semantic metric  $SRQ_k^{(n)}$  to  $s_k^{(n)} = \phi_k \left( SRQ_k^{(n)} \right) \in [0, 1]$ , where  $\phi_k$  is a monotone map fitting the empirical distribution on training solutions. For the higher-is-better metrics (i.e.,  $SRQ_{\text{corr}}$ ,  $SRQ_{\text{diag}}$ ,  $SRQ_{\text{local\_cover}}$ ,  $SRQ_{\text{corr}}^+$ ,  $SRQ_{\text{think}}^+$ , and  $SRQ_{\text{ans}}^+$ ), large values are mapped close to 1; for the lower-is-better metrics ( $SRQ_{\text{think}}^-$  and  $SRQ_{\text{ans}}^-$ ), small values are mapped close to 1; and for the boundary metrics ( $SRQ_{\text{start}}$  and  $SRQ_{\text{end}}$ ),  $\phi_k$  peaks near zero and decreases toward 0 for extreme concentration. Then, we average scores within each group,  $\tilde{s}_{\alpha}^{(n)} = \frac{1}{|K_{\alpha}|} \sum_{k \in K_{\alpha}} s_k^{(n)}$ ,  $\alpha \in \{\text{geo}, \text{sem}\}$ , and define the integrated SRQ score as follows:

$$\widetilde{SRQ}^{(n)} = \tilde{s}_{\text{geo}}^{(n)} + \lambda_{\text{sem}} \tilde{s}_{\text{sem}}^{(n)}. \quad (7)$$

**High and low-SRQ sample selection.** We rank correct solutions  $\mathcal{C}$  and incorrect solutions  $\mathcal{I}$  by  $\widetilde{SRQ}^{(n)}$ . We keep the top 80% of  $\mathcal{C}$  as  $\mathcal{C}_{SRQ^+}$  and the bottom 80% of  $\mathcal{I}$  as  $\mathcal{I}_{SRQ^-}$ , yielding the contrastive groups differing in correctness and self-reading quality.

**Steering direction construction.** Given  $\mathcal{C}_{SRQ^+}$  and  $\mathcal{I}_{SRQ^-}$ , we construct benign self-reading steering directions using standard activation mechanisms. We use the classic CAA (Turner et al., 2023)

as an example. At a chosen layer, we compute the mean token activations  $\mu^{+, (s)}$  and  $\mu^{-, (s)}$  for stage  $s \in \{\text{ans}, \text{reason}\}$  and define

$$v^{(s)} = \mu^{+, (s)} - \mu^{-, (s)}. \quad (8)$$

At inference, we apply  $v^{(\text{ans})}$  to answer-token activations and  $v^{(\text{reason})}$  during reasoning. The SRQ scores can also be combined with other mainstream steering mechanisms such as (Postmus and Abreu, 2024; Højer et al., 2025).

## 5 Experiments and Results

### 5.1 Experimental Setup

**Models.** We employ three frontier thinking LLMs with explicit reasoning traces and short answer stages: R1-Distill-Qwen-7B, R1-Distill-Llama-8B, and Qwen3-4B-Thinking. Decoding for all models uses temperature = 0.65 and top- $p$  = 0.95.

**Datasets.** Our evaluation uses three quantitative reasoning benchmarks: GSM8K (Cobbe et al., 2021), MATH500 (Lightman et al., 2024), and SVAMP (Patel et al., 2021). We construct steering vectors on the training sets of these datasets, where SVAMP is particularly considered for transfer evaluation with steering vectors from GSM8K. All results are reported using accuracy.

**Steering mechanisms.** To assess the compatibility with different steering methods, we instantiate classic CAA (Turner et al., 2023), conceptor steering (Postmus and Abreu, 2024), and recent PCA-CAA (Højer et al., 2025). We follow the standard practice to construct activation directions. See Appendix B for more details about dataset usage, layer selection, and steering construction.

### 5.2 Main Results

We first test whether self-reading signals can serve as a universal supervision source compatible with various steering mechanisms. Table 3 reports the accuracy on GSM8K, MATH500, and SVAMP.

Across all datasets and models, our self-reading steering improves the base models and strengthens existing steering mechanisms. The gains appear consistently for CAA, Conceptor, and PCA-CAA, indicating that SRQ is a generic layer steering the model toward more certain, stable internal states across different mechanisms.

On GSM8K, all three steering mechanisms already bring clear improvements over the non-steered base models. For example, R1-Distill-Qwen-7B rises from 87.8 to 89.1 with PCA-CAA.

Dataset	Thinking LLM	Base	Steering Mechanism			+ Self-Reading Steering		
			CAA	Conceptor	PCA-CAA	+ CAA	+ Conceptor	+ PCA-CAA
GSM8K	R1-Distill-Qwen-7B	87.8	88.4	88.6	89.1	89.5 (1.7 $\uparrow$ )	89.6 (1.8 $\uparrow$ )	<b>89.9 (2.1<math>\uparrow</math>)</b>
	R1-Distill-Llama-8B	85.8	86.7	87.1	87.5	88.1 (2.3 $\uparrow$ )	88.2 (2.4 $\uparrow$ )	<b>88.4 (2.6<math>\uparrow</math>)</b>
	Qwen3-4B-Thinking	91.9	92.2	92.6	92.4	93.0 (1.1 $\uparrow$ )	<b>93.3 (1.4<math>\uparrow</math>)</b>	93.1 (1.2 $\uparrow$ )
MATH500	R1-Distill-Qwen-7B	88.6	89.2	89.6	89.8	90.0 (1.4 $\uparrow$ )	<b>90.4 (1.8<math>\uparrow</math>)</b>	<b>90.4 (1.8<math>\uparrow</math>)</b>
	R1-Distill-Llama-8B	85.6	86.0	86.5	87.1	87.0 (1.4 $\uparrow$ )	87.4 (1.8 $\uparrow$ )	<b>87.9 (2.3<math>\uparrow</math>)</b>
	Qwen3-4B-Thinking	94.2	94.2	94.6	94.4	94.8 (0.6 $\uparrow$ )	<b>95.6 (1.4<math>\uparrow</math>)</b>	95.2 (1.0 $\uparrow$ )
SVAMP	R1-Distill-Qwen-7B	90.3	91.1	91.6	92.0	92.2 (1.9 $\uparrow$ )	92.5 (2.2 $\uparrow$ )	<b>92.7 (2.4<math>\uparrow</math>)</b>
	R1-Distill-Llama-8B	90.7	91.6	92.0	92.4	92.3 (1.6 $\uparrow$ )	92.6 (1.9 $\uparrow$ )	<b>93.2 (2.5<math>\uparrow</math>)</b>
	Qwen3-4B-Thinking	94.5	95.2	95.4	95.5	95.8 (1.3 $\uparrow$ )	96.1 (1.6 $\uparrow$ )	<b>96.5 (2.0<math>\uparrow</math>)</b>

Table 3: Accuracy on GSM8K, MATH500, and SVAMP. The gains of our method over base LLMs are also shown.

LLM	PCA-CAA +			CAA +	
	RePE	SAE-free	Ours	SEAL	Ours
R1-Qwen-7B	89.5	89.6	<b>89.9</b>	88.8	<b>89.5</b>
R1-Llama-8B	87.8	88.1	<b>88.4</b>	87.1	<b>88.1</b>

Table 4: Comparison of steering methods for improving the reasoning trace on GSM8K. The accuracy of the two base models is 87.8 and 85.8.

With our self-reading steering, the accuracy climbs further to 89.9. Qwen3-4B-Thinking is a competing base model with 91.9 accuracy, yet self-reading steering still lifts it to 93.3 with Conceptor. These results indicate that our method yields additive gains by reinforcing a reading mode linked to higher internal certainty, thereby stabilizing LLMs’ cognitive state and improving performance.

MATH500 is a more challenging benchmark with longer solutions, and we see the same trend. Even with long, noisy reasoning traces, guiding the model toward a benign self-reading stabilizes its internal control and monitoring. This facilitates a more decisive process for noisy reasoning contents during the answer stage.

SVAMP assesses how well the learned steering vectors transfer across datasets. Again, self-reading steering yields improvements for all models and mechanisms. The successful transfer to SVAMP confirms that our method targets a general, intrinsic reading strategy rather than overfitting to GSM8K.

### 5.3 Comparison to Other Steering Methods

To validate the effectiveness of self-reading, we compare it with three reasoning-focused steering methods that derive intervention directions from different properties of the reasoning trace: (i) RepE (Højer et al., 2025) steers the model away from chaotic and off-topic behavior toward clearer and

Dataset	LLM	Base	CAA	+ CAA
SciQ	R1-Distill-Qwen-7B	85.9	86.4	<b>86.7</b>
	R1-Distill-Llama-8B	87.3	87.7	<b>88.3</b>
	Qwen3-4B-Thinking	93.9	94.2	<b>94.6</b>
AIME 24–25	R1-Distill-Qwen-7B	46.7	48.3	<b>51.7</b>
	R1-Distill-Llama-8B	41.7	45.0	<b>48.3</b>
	Qwen3-4B-Thinking	76.7	78.3	<b>80.0</b>

Table 5: Generalization to other quantitative reasoning tasks: SciQ and AIME24–25.

LLM	Base	- Geom.	- Sem.	Full
R1-Distill-Llama-8B	85.8	87.3	87.6	<b>88.1</b>
Qwen3-4B-Thinking	91.9	92.5	92.7	<b>93.0</b>

Table 6: Ablation results of R1-Distill-Llama-8B and Qwen3-4B-Thinking on GSM8K. All variants use CAA.

more focused reasoning. (ii) SAE-free (Li et al., 2025) analyzes reasoning traces and extracts directions that emphasize mathematical and logical structure while suppressing generic narrative content. (iii) SEAL (Chen et al., 2025) promotes concise and efficient reasoning by preferring execution-like representations and reducing reflection and transition. Following the literature, RepE and SAE-free use PCA-CAA, while SEAL uses CAA.

Table 4 presents the accuracy comparison on GSM8K with R1-Distill-Qwen-7B and R1-Distill-Llama-8B. All steering methods improve the non-steered base models. Based upon the PCA-CAA mechanism, our self-reading steering reaches 89.9 and 88.4, surpassing both RepE and SAE-free. Similarly, with the CAA mechanism, our method outperforms SEAL. The consistency of improvements highlights the robustness of our method. These results indicate that steering the model toward internally certain states characterized by benign self-

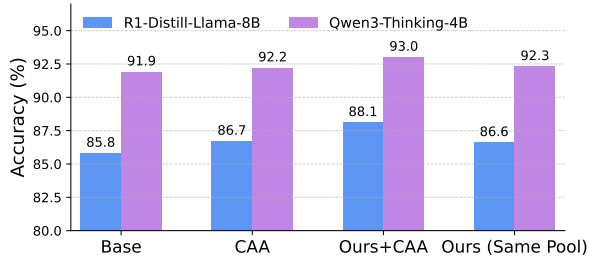


Figure 3: Comparison on GSM8K with CAA. Instead of using a larger candidate pool to obtain 700 pairs, the “Same Pool” variant draws from the same pool sampled by CAA and filters it to retain a smaller set.

reading patterns is a meaningful and effective objective. The self-reading signals facilitate the selection of higher-quality samples for intervention, thereby leading to better performance.

#### 5.4 Generalization to Broader Tasks

To further explore the generalization across quantitative reasoning tasks, we evaluate our self-reading steering method on SciQ (Scientific Question Answering) (Welbl et al., 2017) and AIME24–25 (Mathematical Association of America, 2025) (American Invitational Mathematics Examination). These datasets require more complex reasoning chains (Chen et al., 2024). We construct vectors from SciQ’s training split and reuse MATH500 vectors for AIME. As shown in Table 5, our method consistently improves the base models and the CAA mechanism. Particularly, on the very difficult AIME24–25, the improvement reaches up to 6.6 points over the base model R1-Distill-Llama-8B, which corresponds to four additional competition problems solved. The success on these tasks confirms that our method filters a general signal of internal certainty.

#### 5.5 Ablation Study on SRQ Dimensions

We conduct an ablation study to investigate the individual contributions of the geometric and semantic dimensions of SRQ scores. As shown in Table 6, removing either dimension reduces performance, with the geometric component having a stronger impact on steering effectiveness. Through manual inspection of samples, we find that the good geometric features corresponding to forward-shifting attention may have already captured the key steps in these traces.

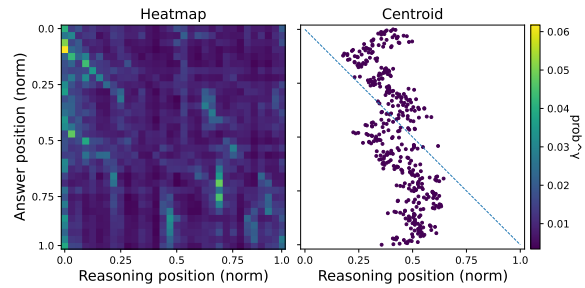


Figure 4: Visualization of a special self-reading error in R1-Distill-Llama-8B.

#### 5.6 Analysis of Candidate Set Fairness

To isolate the benefit of SRQ selection from the potential impact of a larger candidate pool, we conduct a “Same Pool” analysis. We apply SRQ scores to the pairs of correct and incorrect traces used by the standard steering mechanism and filter them to retain a subset of pairs. As presented in Figure 3, the “Same Pool” variant remains competitive with or slightly outperforms the baseline while using fewer samples to construct the steering vectors. This verifies that activation steering vectors extracted from the smaller subset of samples by SRQ remain highly effective.

#### 5.7 Case Study of Uncertain Self-Reading

Incorrect answers often exhibit irregular attention without a consistent organization. Figure 4 highlights a special counter case where the reasoning trace reaches the correct result but is long, cluttered, and filled with reflective “waits”, yet the answer stage is wrong.

The heatmap shows dark colors without clear oblique bands or anchor columns, indicating broadly distributed attention over the reasoning trace. Isolated bright spots near the problem statement remain inconsistent with the benign pattern. The centroid plot shows that rather than a smooth forward drift, the centroids form a nearly vertical broken line that oscillates within the same region. This suggests a failure of both control and monitoring: the model does not maintain a certain solution branch or lock onto semantic anchors during answer decoding. The model wanders within a noisy trace and fails to consolidate information for a correct prediction. More attention visualizations of general failure cases are detailed in Appendix A.3.

LLM	Base		Ours + CAA	
	Acc	Conf	Acc	Conf
R1-Distill-Llama-8B	85.8	82.8%	88.1	84.1%

Table 7: Comparison of accuracy and confidence on GSM8K.

## 5.8 Analysis of Benign Self-reading and Cognitive Certainty

We hypothesize that facilitating benign reading patterns induces a state of higher cognitive certainty, thereby enhancing reasoning performance. To verify this, we compare the model’s inference confidence with and without the intervention of SRQ activation vectors. We quantify the model’s overall certainty by computing the mean confidence over the complete generation process. The confidence is defined as follows:

$$Conf = \left( \prod_{i=1}^n \max_{a_t \in V} p(a_t) \right)^{\frac{1}{n}}, \quad (9)$$

where  $p(a_t) = \Pr[a_t | H, I, a_{<t}]$  calculates the model’s predicted probability for the answer token  $a_t$ . Here,  $H$  denotes the context including the input prompt and generated thoughts,  $I$  is the instruction to elicit the answer,  $n$  is the number of decoding steps,  $V$  is the vocabulary, and  $a_{<t}$  denotes all previously generated tokens (Liu et al., 2025; Yang et al., 2025b).

As presented in Table 7, guiding the model with the activation vector leads to consistent improvements in both accuracy and confidence, confirming that our method effectively boosts the model’s internal certainty.

## 5.9 Scaling Analysis

We extend our evaluation to larger models to verify whether the effectiveness of our approach persists as model capacity increases. Table 8 lists that our self-reading steering consistently yields improvements on both the 7B and 14B models. Although the baseline performance of R1-Distill-Qwen-14B is already strong, our method still brings modest gains. These results demonstrate that the signals captured by SRQ remain robust and effective across different model scales.

## 6 Conclusion

We study how answer tokens read the reasoning trace in thinking LLMs for quantitative reasoning,

Dataset	LLM	Base	CAA	+ CAA
		GSM8K	R1-Distill-Qwen-7B	87.8
	R1-Distill-Qwen-14B	94.6	94.9	<b>95.2</b>
MATH500	R1-Distill-Qwen-7B	88.6	89.2	90.0
	R1-Distill-Qwen-14B	93.2	93.6	<b>93.9</b>

Table 8: Scaling results on GSM8K and MATH500.

and discover a benign self-reading pattern with forward-shifting attention centroids and repeated focus on semantic anchors that is strongly associated with model certainty. We propose a steering method driven by SRQ scores combining geometric and semantic metrics, which promotes certain internal states and well-organized self-reading at inference time. Our experiments show consistent accuracy gain, suggesting self-reading is an effective and generalizable signal, and SRQ is likely to remain useful as steering methods evolve.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62272219) and the CCF-1688 Yuanbao Cooperation Fund (No. CCF-Alibaba2025001).

## Ethical Considerations

The datasets and benchmarks used in this work are publicly available and distributed under permissive licenses, and we follow their stated terms of use. For result analysis, we recruited graduate students to perform manual annotations, and informed consent was obtained from all participants. Apart from these annotations, our work does not involve collecting new human data, personal or sensitive information, or interventions in real-world user-facing systems. Nevertheless, as with any work on LLMs, our methods could inherit biases or limitations present in the underlying models or datasets, and results should be interpreted with this in mind.

## Limitations

First, our study focuses on thinking LLMs that expose an explicit reasoning trace followed by a dedicated answer stage, and our SRQ metrics are defined over answer-to-reasoning attention and intermediate activations. This limits direct applicability to open-source thinking LLMs where such internal signals are accessible, and the best layer choices may vary across architectures.

Second, semantic SRQ relies on span-level annotations produced by external LLMs, which introduces additional cost and potential label noise. Developing cheaper and fully automatic proxies remains a future work. Also, the benefits of steering the answer stage are bounded by the quality of the available evidence in the reasoning trace. If the trace is fundamentally incorrect or lacks key steps, self-reading alone may have limited improvement headroom, motivating tighter coupling between reasoning-stage and answer-stage interventions.

Finally, our analysis is centered on quantitative reasoning, where the benign self-reading pattern appears most consistently, especially on math-style problems. For other domains, it remains unclear whether they exhibit the same or different self-reading patterns. Understanding how domain characteristics relate to self-reading behaviors is an important direction of future work.

## References

- Riccardo Ali, Francesco Caso, Christopher Irwin, and Pietro Liò. 2025. *Entropy-Lens: The information signature of transformer computations*. *CoRR*.
- Seyedarmin Azizi, Erfan Baghaei Potraghloo, and Masoud Pedram. 2025. *Activation steering for chain-of-thought compression*. *CoRR*.
- Runjin Chen, Zhenyu Zhang, Junyuan Hong, Souvik Kundu, and Zhangyang Wang. 2025. *SEAL: Steerable reasoning calibration of large language models for free*. *CoRR*.
- Songlin Chen, Weicheng Wang, Xiaoliang Chen, Peng Lu, Zaiyan Yang, and Yajun Du. 2024. *LLaMA-LoRA neural prompt engineering: A deep tuning framework for automatically generating Chinese text logical reasoning thinking chains*. *Data Intelligence*, 6(2):375–408.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. *Training verifiers to solve math word problems*. *CoRR*.
- DeepSeek-AI. 2025. *DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning*. *CoRR*.
- Leon Eshuijs, Archie Chaudhury, Alan McBeth, and Ethan Nguyen. 2025. *But what is your honest answer? Aiding LLM-judges with honest alternatives using steering vectors*. *CoRR*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. *Measuring mathematical problem solving with the MATH dataset*. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.
- Bertram Højer, Oliver Simon Jarvis, and Stefan Heinrich. 2025. *Improving reasoning performance in large language models via representation engineering*. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Asher Koriat. 1997. *Monitoring one’s own knowledge during study: A cue-utilization approach to judgments of learning*. *Journal of experimental psychology: General*, 126(4):349.
- Anna Langedijk, Hosein Mohebbi, Gabriele Sarti, Willem H. Zuidema, and Jaap Jumelet. 2024. *DecoderLens: Layerwise interpretation of encoder-decoder transformers*. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 4764–4780. Association for Computational Linguistics.
- Zihao Li, Xu Wang, Yuzhe Yang, Ziyu Yao, Haoyi Xiong, and Mengnan Du. 2025. *Feature extraction and steering for enhanced chain-of-thought reasoning in language models*. *CoRR*.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. *Let’s verify step by step*. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Yi Liu, Xiangyu Liu, Zequn Sun, and Wei Hu. 2025. *Answering the unanswerable is to err knowingly: Analyzing and mitigating abstention failures in large reasoning models*. *CoRR*.
- Mathematical Association of America. 2025. *American invitational mathematics examination (AIME)*. <https://maa.org/maa-invitational-competitions/>. Accessed January 5, 2026.
- Thomas O Nelson. 1990. *Metamemory: A theoretical framework and new findings*. In *Psychology of learning and motivation*, volume 26, pages 125–173. Elsevier.
- Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. *Are NLP models really able to solve simple math word problems?* In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2080–2094. Association for Computational Linguistics.

- Joris Postmus and Steven Abreu. 2024. [Steering large language models using conceptors: Improving addition-based activation engineering](#). *CoRR*.
- Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. 2025. [Layer by layer: Uncovering hidden representations in language models](#). In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.
- Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. 2025. [Improving instruction-following in language models through activation steering](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Chung-En Sun, Ge Yan, and Tsui-Wei Weng. 2025. [ThinkEdit: Interpretable weight editing to mitigate overly short thinking in reasoning models](#). *CoRR*.
- Ala N. Tak, Amin Banayeezade, Anahita Bolourani, Mina Kian, Robin Jia, and Jonathan Gratch. 2025. [Mechanistic interpretability of emotion inference in large language models](#). In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 13090–13120. Association for Computational Linguistics.
- Yiru Tang, Kun Zhou, Yingqian Min, Wayne Xin Zhao, Jing Sha, Zhichao Sheng, and Shijin Wang. 2025. [Enhancing chain-of-thought reasoning via neuron activation differential analysis](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16162–16170.
- Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. [Steering language models with activation engineering](#). *CoRR*.
- Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. 2025. [Understanding reasoning in thinking language models via steering vectors](#). *CoRR*.
- Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. [Crowdsourcing multiple choice science questions](#). In *Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 94–106. Association for Computational Linguistics.
- Tianhao Wu, Janice Lan, Weizhe Yuan, Jiantao Jiao, Jason E. Weston, and Sainbayar Sukhbaatar. 2025. [Thinking LLMs: General instruction following with thought generation](#). In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 40 others. 2025a. [Qwen3 technical report](#). *CoRR*.
- Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Zheng Lin, Li Cao, and Weiping Wang. 2025b. [Dynamic early exit in reasoning models](#). *CoRR*.
- Jue Zhang, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2025. [From reasoning to answer: Empirical, attention-based and mechanistic insights into distilled DeepSeek R1 models](#). *CoRR*.
- Yu Zhao, Alessio Devoto, Giwon Hong, Xiaotang Du, Aryo Pradipta Gema, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. 2025a. [Steering knowledge selection behaviours in LLMs via SAE-based representation engineering](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 5117–5136. Association for Computational Linguistics.
- Zekai Zhao, Qi Liu, Kun Zhou, Zihan Liu, Yifei Shao, Zhiting Hu, and Biwei Huang. 2025b. [Activation control for efficiently eliciting long chain-of-thought ability of language models](#). *CoRR*.

## A More Self-Reading Analysis

### A.1 Self-Reading on Math500

To test the robustness of benign self-reading on harder quantitative reasoning tasks, we repeat the analysis on Math500 (Lightman et al., 2024). As illustrated in Figure 5, the aggregated results from 40 correct solutions show a broad diagonal attention band drifting rightward, confirming that the forward-shifting reading focus persists in longer reasoning traces. The band is less regular than that in GSM8K and is interwoven with weaker slanted streaks, reflecting longer traces and more reflective segments where the model briefly revisits. This suggests that the same forward moving reading strategy remains active even when the problems require more extended reasoning.

We also visualize several *single-instance* answer-to-reasoning attention heatmaps and centroid plots for all three models on Math500. We use the question depicted in Figure 6. Here, we reiterate the concept of the attention centroid. The attention centroid represents the weighted average position where an answer token “focuses” within the reasoning sequence. Specifically, for each answer token, we treat its row-normalized attention weights as

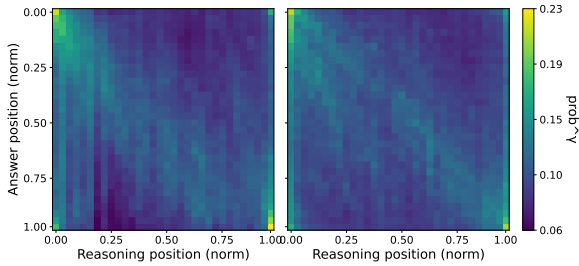


Figure 5: The aggregated heatmaps with Qwen3-4B-Thinking (left) and R1-Distill-Llama-8B (right) on the Math500 benchmark.

"The sum of the first  $n$  terms in the infinite geometric sequence  $\left\{\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \dots\right\}$  is  $\frac{255}{512}$ . Find  $n$ ."

Figure 6: The question in Math500 for visualization.

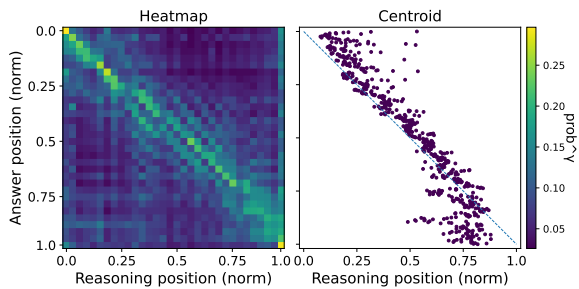


Figure 7: Single-instance answer-to-reasoning attention of R1-Distill-Qwen-7B.

probabilities, multiply each reasoning token’s position index by its attention weight, and sum these weighted positions to obtain the centroid, which is then normalized to  $[0,1]$  for comparison across sequences of different lengths.

The visualizations for R1-Distill-Qwen-7B (layer 21), R1-Distill-Llama-8B (layer 22), and Qwen3-4B-Thinking (layer 23) are shown in Figures 7–9, respectively. Compared with GSM8K, the Math500 solutions typically involve longer reasoning traces with more reflective detours, making the heatmaps and centroid plots less regular. Nevertheless, the benign self-reading pattern characterized by the two features remains consistent. As decoding proceeds, the answer-token attention centroid shifts steadily from earlier to later parts of the reasoning trace, while repeatedly revisiting key semantic anchors for verification.

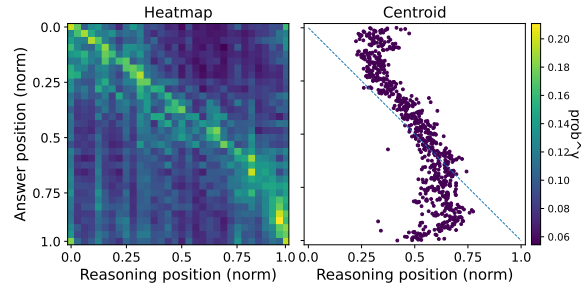


Figure 8: Single-instance answer-to-reasoning attention of R1-Distill-Llama-8B.

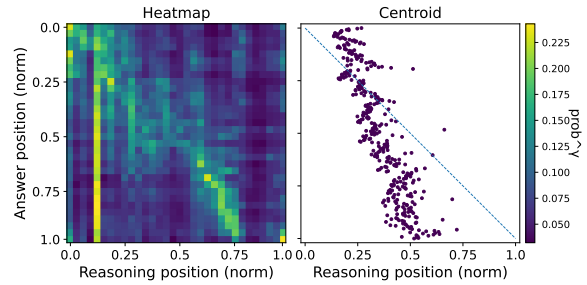


Figure 9: Single-instance answer-to-reasoning attention of Qwen3-4B-Thinking.

## A.2 Uncertain Self-Reading Case

Figure 10 presents the full text output of the uncertain self-reading case in Section 5.7. In this example, the reasoning (the content before  $\langle /think \rangle$ ) ultimately reaches the correct conclusion, while the answer (after  $\langle /think \rangle$ ) is incorrect.

As shown in the figure, the reasoning is long and includes multiple “Wait”. The key issue is an early misreading of the problem, which triggers misguided branches of reasoning. Although the model eventually corrects itself and arrives at the right result, the overall reason is low-quality, messy, inconsistent, and filled with “Wait”. This serves as a clear counter-example to a healthy reading pattern: when the reason becomes noisy and contains erroneous branches, the answer stage requires stable self-reading to stay consistent. Guiding the model toward benign reading mode and a more certain internal state should therefore increase the likelihood of a correct final answer.

## A.3 General Unstructured Failure Cases

Figure 10 shows an informative counter-example, yet such cases are rare. The answer-to-reasoning attention in most incorrect GSM8K generations becomes irregular and fragmented rather than forming a coherent reading trajectory. Figure 11 illustrates an instance of incorrect response to a specific

Reasoning Content

First, maybe I should figure out how many red and blue cubes Grady gave to Gage. Since Grady has 20 red cubes initially, and Gage has 10 red after receiving some, that means Grady gave away 10 red cubes, right? Because 20 minus 10 is 10. Hmm, let me write that down. Red cubes given:  $20 - 10 = 10$  red cubes. Okay, and for the blue cubes, Grady started with 15. Gage has 12 blue cubes now. So Grady must have given away 3 blue cubes because  $15 - 12 = 3$ . Let me note that. Blue cubes given:  $15 - 12 = 3$  blue cubes. Wait, the problem says he gives  $2/5$  of his red numbered cubes and  $1/3$  of the blue numbered cubes. So I should check if 10 red is  $2/5$  of 20, and 3 blue is  $1/3$  of 15. Let me calculate  $2/5$  of 20. 20 divided by 5 is 4, so 2 times 4 is 8. Wait, that's not 10. Hmm, that doesn't add up. Maybe I did something wrong. Wait, hold on. Maybe I need to set up equations. Let's let  $x$  be the number of red cubes Grady gives, and  $y$  be the number of blue cubes he gives. According to the problem,  $x = 2/5 * 20$ . Let me compute that:  $2/5$  of 20 is 8. So he gives 8 red cubes to Gage. Similarly,  $y = 1/3 * 15$ , which is 5. So he gives 5 blue cubes to Gage. Wait, but earlier I thought he gave 10 red and 3 blue. There's a discrepancy here. Hmm. Wait, maybe I misread the problem. Let me read it again. "Grady has 20 red numbered cubes and 15 blue numbered cubes. He gives his friend Gage  $2/5$  of his red numbered cubes and  $1/3$  of the blue numbered cubes. If Gage had 10 red numbered cubes and 12 blue numbered cubes, find the total number of cubes Gage has?" "Oh! Wait, maybe the problem is that Grady gives Gage  $2/5$  of his red cubes and  $1/3$  of his blue cubes, and after that, Gage has 10 red and 12 blue. So maybe Gage initially had some cubes, but the way it's phrased, it seems Grady gives him some cubes, and then Gage ends up with 10 red and 12 blue. So maybe Gage started with zero? Or maybe he had some before? Wait, if Gage had 10 red numbered cubes and 12 blue numbered cubes' might be telling me what he already had before getting anything from Grady. If that's true, then I shouldn't be subtracting from 20 and 15 at all. He receives 8 red and 5 blue, so he would end up with  $10 + 8 = 18$  red and  $12 + 5 = 17$  blue. Wait, but the problem says "He gives his friend Gage  $2/5$  of his red numbered cubes and  $1/3$  of the blue numbered cubes. If Gage had 10 red numbered cubes and 12 blue numbered cubes..." Hmm, it might be implying that after giving those amounts, Gage has 10 red and 12 blue. So perhaps Gage didn't have any before. So he had 0, and Grady gave him some. Wait, but the wording is a bit ambiguous. It says "He gives his friend Gage  $2/5$  of his red numbered cubes and  $1/3$  of the blue numbered cubes. If Gage had 10 red numbered cubes and 12 blue numbered cubes..." "Hmm, maybe it's a translation issue or wording issue. Maybe Gage already had some cubes, and Grady gives him  $2/5$  of his red and  $1/3$  of his blue, resulting in Gage having 10 red and 12 blue. So Gage had some before. So, let me model it as: Let  $G_{red}$  be the number of red cubes Gage had before receiving from Grady. Let  $G_{blue}$  be the number of blue cubes Gage had before receiving. After Grady gives him  $2/5$  of his red and  $1/3$  of his blue, Gage has 10 red and 12 blue. So, Gage's red after receiving is  $G_{red} + (2/5)(20) = 10$ . Similarly, Gage's blue after receiving is  $G_{blue} + (1/3)(15) = 12$ . So, set up the equations:  $G_{red} + (2/5)*20 = 10$  and  $G_{blue} + (1/3)*15 = 12$ . Solve for  $G_{red}$  and  $G_{blue}$ . So,  $G_{red} + 8 = 10$ . Therefore,  $G_{red} = 2$ .  $G_{blue} + 5 = 12$ . Therefore,  $G_{blue} = 7$ . So, before receiving any cubes, Gage had 2 red and 7 blue. Then, after receiving, he has  $2 + 8 = 10$  red and  $7 + 5 = 12$  blue. Therefore, total cubes Gage has is  $10 + 12 = 22$ . Wait, but the question is asking for the total number of cubes Gage has. So  $10 + 12$  is 22. Wait, but let me double-check. Grady has 20 red and 15 blue. He gives  $2/5$  of red:  $2/5 * 20 = 8$  red to Gage. He gives  $1/3$  of blue:  $1/3 * 15 = 5$  blue to Gage. So Grady gives away 8 red and 5 blue, so Grady is left with 12 red and 10 blue. But the problem says Gage had 10 red and 12 blue. So if Grady gave away 8 red and 5 blue, that would mean Gage had  $10 - 8 = 2$  red before? Wait, no, that doesn't make sense. Wait, maybe I need to think differently. Wait, maybe Grady gave Gage  $2/5$  of his red and  $1/3$  of his blue, and Gage had 10 red and 12 blue after that. So, to find out how many cubes Gage had before, we can subtract the given fractions from Gage's current cubes. So, Gage had 10 red. He received  $2/5$  of Grady's red, which is 8, so he had  $10 - 8 = 2$  red before. Similarly, Gage had 12 blue. He received  $1/3$  of Grady's blue, which is 5, so he had  $12 - 5 = 7$  blue before. Therefore, Gage had  $2 + 7 = 9$  cubes before. But wait, the problem is asking for the total number of cubes Gage has, which is 10 red and 12 blue, so 22 in total. So, the answer is 22. Wait, but the problem is phrased as Grady gives Gage some cubes, so I think Gage's total is 10 red and 12 blue, so 22 in total. But let me think again. If Grady gave 8 red and 5 blue to Gage, then Grady would have 12 red and 10 blue left. But the problem doesn't say that. Alternatively, maybe the problem is simply asking how many does Gage have, which is  $10 + 12 = 22$ . So, I think 22 is the answer. So, in total, Gage has 22 cubes. Wait, but to be thorough, let me just check. If Gage had 2 red and 7 blue before, Grady gave him 8 red and 5 blue, which adds up to 9 cubes. Then, Gage now has  $2 + 8 = 10$  red and  $7 + 5 = 12$  blue, which is 22 in total. Wait, if the answer were just  $10 + 12$ , then the fractions  $(2/5)$  and  $(1/3)$  didn't really change the final total at all. That makes the setup feel pointless. So, the wording "If Gage had 10 red numbered cubes and 12 blue numbered cubes' sounds more like a starting condition than a final result. So I should add what Grady gave him:  $10 + 8 = 18$  red and  $12 + 5 = 17$  blue, for a total of 35. Wait, it could still just be 22 if the '10 red and 12 blue' are already the after numbers. The question might simply be asking for the total of those two numbers. Wait, that makes the fractions useless. The problem probably wants me to use all the steps. So,  $18 + 17 = 35$  seems like the intended answer. So, yes, that makes sense. So I'll go with 35 cubes in total. \*\*Final Answer\*\* The total number of cubes Gage should be 35, rather than 22. </think>

Answer Content

Grady has 20 red numbered cubes and 15 blue numbered cubes. He gives his friend Gage  $2/5$  of his red numbered cubes and  $1/3$  of the blue numbered cubes. After giving these cubes, Gage has 10 red numbered cubes and 12 blue numbered cubes. First, we calculate the number of red and blue cubes Grady gives to Gage: Grady gives Gage  $(\frac{2}{5})$  of his 20 red cubes:  $(\frac{2}{5}) \times 20 = 8$  red cubes. Grady gives Gage  $(\frac{1}{3})$  of his 15 blue cubes:  $(\frac{1}{3}) \times 15 = 5$  blue cubes. Next, we determine the total number of cubes Gage has after receiving these cubes: Gage has 10 red cubes and 12 blue cubes. The total number of cubes Gage has is:  $10 + 12 = 22$ . However, considering the problem's context and the fractions given, it is more consistent to calculate the total number of cubes Gage has by adding the cubes he received from Grady to his current count. Therefore, the correct total number of cubes Gage has is:  $10 + 12 = 22$ . But upon re-evaluation, if we consider the cubes Grady gave to Gage and the current count, the total number of cubes Gage has is:  $10 + 12 = 22$ . However, the problem might be interpreted as Gage having 10 red and 12 blue cubes after receiving the cubes from Grady. Therefore, the total number of cubes Gage has is:  $10 + 12 = 22$ . But considering the initial interpretation and the fractions given, the correct answer should be: 35. But after careful consideration, the correct total number of cubes Gage has is: 22.

Figure 10: Example of uncertain self-reading.

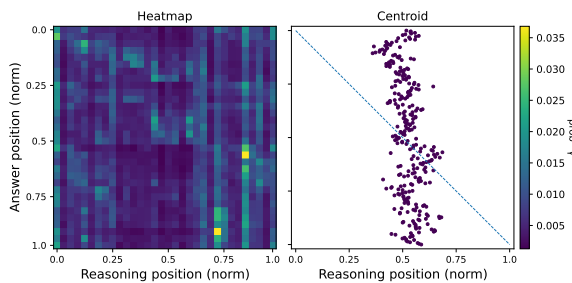


Figure 11: Failure case on GSM8K.

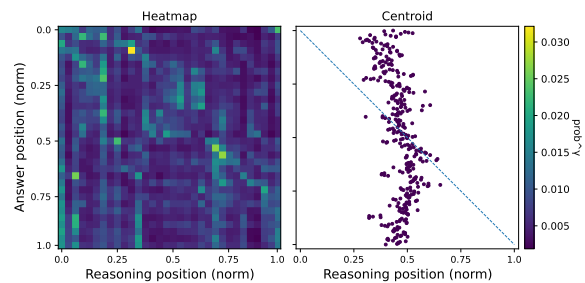


Figure 12: Failure case on MATH500.

GSM8K problem, while Figure 12 shows a failure example of a Math500 problem, both from the R1-Distill-Llama-8B model.

The GSM8K question is: “Carlos is planting a lemon tree. The tree will cost 90 to plant. Each year it will grow 7 lemons, which he can sell for 1.5 each. It costs 3 a year to water and feed the tree. How many years will it take before he starts earning money on the lemon tree?”

The Math500 question is “An investment of \$24,000 is made in a government bond that will pay 1% bi-monthly interest (meaning that the investment will increase by 1% every two months). At the end of five years, what is the total number of dollars in this investment? \n\n Express your answer to the nearest whole number.”.

These heatmaps exhibit scattered, noisy high-attention regions, and two centroid plots curve collapse into near-vertical traces, indicating no forward drift over the reasoning. Unlike benign self-reading, the model does not appear to commit to a consistent solution branch and read it through in order, which is consistent with recent evidence (Ali et al., 2025) that less structured attention signals correlate with lower answer correctness. Additionally, a very small fraction of incorrect samples exhibit benign self-reading, which is why our self-reading steering excludes the top 20% highest-SRQ solutions from the incorrect set.

## B Details of Experiment Setup

### B.1 Model and Layer Selection

We study three thinking LLMs with decoupled reasoning and answer stages: R1-Distill-Qwen-7B, R1-Distill-Llama-8B, and Qwen3-4B-Thinking. They all produce an explicit chain-of-thought style reasoning trace followed by a short final answer segment, which makes them well suited for self-reading analysis. For  $SRQ_{\text{local\_cover}}$ , we set  $\beta = 0.7$ ,  $\tau = 0.4$ , and  $w$  to 20% of the length of the reasoning trace. For  $SRQ_{\text{think}}$ , we set  $\gamma = 0.1$ . For  $SRQ_{\text{start/end}}$ , we set  $\rho_{\text{bd}} = 0.05$ , and  $\alpha = 1.6$  for  $\rho_{\text{tar}}$ . For  $\widetilde{SRQ}^{(n)}$ , we simply set  $\lambda_{\text{sem}} = 1$ . For all experiments, we use a consistent decoding configuration with temperature = 0.65 and top- $p = 0.95$  to minimize confounding factors from sampling.

Our self-reading steering operates at the activation level of intermediate transformer layers. Prior analytical studies and activation engineering works (Skean et al., 2025; Langedijk et al., 2024; Tak et al., 2025; Zhao et al., 2025a) indicate that mid-

to-late layers encode richer semantic information about the ongoing reasoning process while being less sensitive to superficial lexical variations, making them a favorable choice for activation-level interventions. Consequently, steering at these layers yields more reliable modulation of reasoning behavior. We therefore perform preliminary runs across several candidate layers and select those providing a good trade-off between stability and steering strength. In the experiments, we use layer 21 for R1-Distill-Qwen-7B, layer 20 for R1-Distill-Llama-8B, and layer 22 for Qwen3-4B-Thinking.

### B.2 Datasets

We conduct evaluation on GSM8K (Cobbe et al., 2021), MATH500 (Lightman et al., 2024), and SVAMP (Patel et al., 2021).

GSM8K is a collection of grade-school level math word problems. We follow the standard split and use the training split to run the models and record complete solutions as well as internal activations, while reserving the official test split exclusively for evaluation.

MATH500 is a subset of the MATH dataset (Hendrycks et al., 2021), which contains more challenging competition-style math problems. For MATH500, we use the full MATH training set to obtain reasoning traces and activations for building steering vectors, and evaluate steering performance on the 500-problem subset.

SVAMP is a challenge set of arithmetic word problems designed to probe robustness to superficial variations. In our experiments, SVAMP is used purely as an evaluation benchmark for *forward transfer*. Specifically, we do not collect separate steering data on SVAMP. Instead, we reuse steering vectors derived from the GSM8K activations and apply them directly to SVAMP, allowing us to test whether the SRQ-guided self-reading signals generalize across related quantitative reasoning distributions.

### B.3 Steering Mechanisms

To probe the generality of SRQ-based self-reading steering w.r.t. the underlying representation engineering method, we instantiate three steering mechanisms: CAA (Turner et al., 2023), Conceptor-based steering (Postmus and Abreu, 2024), and PCA-CAA (Højer et al., 2025). CAA performs class-wise activation addition by constructing a direction from mean activations of contrastive positive and negative sample sets. Conceptor-based

steering first uses conceptor matrices to capture subspaces associated with desired and undesired behaviors and then modifies activations by projecting along or away from these subspaces. PCA-CAA first applies principal component analysis to the collected activations and then performs class-wise activation addition in the resulting low-dimensional space, which helps separate components that are relevant to reasoning from those that are not.

For all three steering mechanisms, we follow standard practice in constructing supervision signals. Correct solutions are treated as positive samples and incorrect solutions as negative samples. For all three methods, we first obtain the activation representation of each positive or negative sample at the selected layer by averaging the hidden states over all tokens in its generated solution. The resulting per-sample mean vectors are then used to construct the corresponding steering directions. In our baseline steering construction, activations are aggregated over all tokens in the generated solution and applied uniformly to every token during decoding. In contrast, our self-reading steering, driven by SRQ scores, further filters correct and incorrect samples and separately extracts and steers activations for the reasoning and answer segments.

Except for the experiments in Section 5.6, we use 700 positive and 700 negative samples to construct steering vectors for each model–dataset pair, so that the final amount of data used to compute activation vectors is matched across methods. For the baseline methods, including CAA, Conceptor, and PCA-CAA, we directly and randomly select 700 correct and 700 incorrect solutions.

For our method, the final steering set also contains 700 positive and 700 negative samples, but the initial candidate pool is larger. Specifically, we first collect 875 correct and 875 incorrect solutions, and then apply SRQ-based filtering: we retain the top 80% of correct solutions ranked by SRQ and the bottom 80% of incorrect solutions, which yields the final 700+700 traces used for steering vector construction. In this way, our method keeps the final sample size the same as the baselines, while improving the quality of the selected traces.

To further isolate the effect of SRQ from the use of a larger initial candidate pool, Section 5.6 reports a *Same Pool* fairness experiment. There, SRQ-based filtering is applied to exactly the same 700+700 candidate traces used by the baseline methods, retaining only 560+560 traces after filtering, yet still outperforming the baselines. This

Collected Cases	Remain Correct	Flip to Incorrect
100	97	3

Table 9: 100 GSM8K cases from DeepSeek-R1-Distill-Llama-8B in which the model initially reaches the correct answer through explicit reflective behaviors.

result suggests that the gains mainly come from SRQ-based sample selection rather than simply from starting with a larger pool.

For Qwen3-4B-Thinking, its high accuracy on GSM8K makes it challenging to obtain sufficient incorrect solutions directly from the model. To alleviate this, we additionally construct 200 incorrect answers using GPT-5, which serve as extra negative samples when building steering directions for Qwen3-4B-Thinking.

## C Preservation of Reflection under SRQ Steering

To verify that SRQ steering improves the stability of answer generation without suppressing the model’s ability to perform reflective self-correction, we conduct an additional analysis on DeepSeek-R1-Distill-Llama-8B. The goal of SRQ is to promote benign self-reading at the answer stage and to select higher-quality reasoning traces and answer states. In particular, the semantic dimension of SRQ (Section 4.3) emphasizes attention concentration on key semantic anchors, including reflective phrases such as “wait,” “let me check,” and similar back-references. Therefore, when the model engages in goal-directed reflection, such behaviors typically contribute positively to SRQ.

First, we manually collected 100 GSM8K cases from DeepSeek-R1-Distill-Llama-8B in which the model arrived at the correct answer through explicit reflective behaviors, such as self-checking or reconsideration. As shown in Table 9, after applying SRQ+CAA, 97 of these cases remained correct, and only 3 flipped to incorrect, indicating that corrective reflection is largely preserved under steering. Second, Table 10 shows that reflective markers remain common after steering. On the one hand, the average number of occurrences of “wait” per sample shows only a slight decrease; on the other hand, the average occurrence of “check” remains nearly unchanged. This pattern suggests that SRQ steering does not eliminate reflection itself. Instead, it appears to reduce redundant or disorganized hesitation while preserving deliberate verification be-

Marker	Total		Avg./Sample	
	Base	SRQ+CAA	Base	SRQ+CAA
wait	12393	10066	9.40	7.63
check	1706	1715	1.29	1.30

Table 10: Lexical statistics of reflective markers on GSM8K for DeepSeek-R1-Distill-Llama-8B.

Dataset	Model	Base	SRQ+CAA	$\Delta$
GSM8K	R1-Llama-8B	5801	5749	-52
	Qwen3-4B	3012	2750	-262
MATH500	R1-Llama-8B	10381	10358	-23
	Qwen3-4B	13673	13325	-348

Table 11: Comparison of Average Generation Length (in Characters).

haviors that support correct answering. Together with the accuracy gains on challenging benchmarks such as MATH500 and AIME, these results suggest that SRQ steering improves answer-stage stability without materially affecting the model’s capacity for reflective self-correction.

## D Effect on Generation Length

As discussed in Section C, our method does not suppress the model’s ability to reflect and self-correct. At the same time, since SRQ encourages the model to adopt a more confident and stable self-reading pattern during generation, it may incidentally reduce unnecessary wandering or verbosity. We therefore also examine whether SRQ-steering affects the length of generated traces. Specifically, we measure the average generation length after applying SRQ+CAA. As shown in Table 11, the change in output length is small across all settings, and in most cases the steered model produces slightly shorter traces than the base model. This suggests that SRQ-steering has only a minor overall effect on output length, although there is indeed a slight tendency toward shorter generations.

## E Isolation Analysis of Reading Phase

To further isolate the importance of benign self-reading at the answer stage, we perform an analysis that applies steering only during answer decoding, thereby excluding the typically required steering of the reasoning trace.

Figure 13 shows that even under this deliberately constrained setting, answer-only steering outperforms the non-steered baseline and remains com-

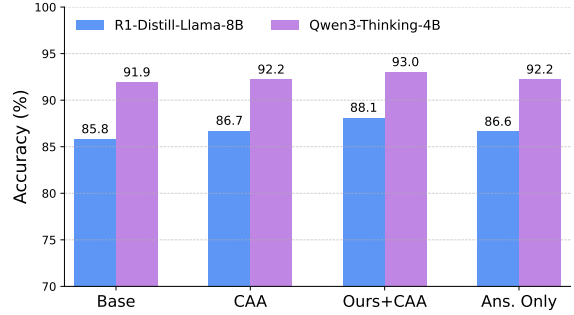


Figure 13: Clean steering on GSM8K: steering is applied only during answer decoding.

```
"This is a math problem.\n"
"Please first write out your complete thought process\n"
"within the <think> and </think> tags, and then provide\n"
"your answer."\n"
"Control the length of your thinking—don't make it too\n"
"long.\n\n"
f"Question: \n{question}\n"
```

Figure 14: Prompt used for self-reading analysis.

petitive with CAA. For instance, R1-Distill-Llama-8B reaches 86.6, which is nearly identical to CAA (86.7) and above the base model (85.8).

The weaker gains compared to full steering are expected, because this isolation leaves errors within the reasoning trace uncorrected, removing this standard steering channel. If the trace is flawed, improving how the answer reads the trace has limited headroom as there is no correct evidence to use. This analysis shows that performance gains achieved without steering the reasoning trace come from better utilization of the already available evidence. This addresses the specific failure mode where correct reasoning yields unfaithful answers. Although such mismatches are rare in our manual inspection, the observed performance improvement supports that self-reading quality at the answer stage represents a form of internal model certainty, which is a critical factor affecting final correctness.

## F Relevant Prompts

We briefly summarize the prompts used in our work for model solution generation and automatic answer evaluation.

For the self-reading analysis, we use a math-oriented generation prompt shown in Figure 14. The prompt instructs the model to first write out its complete thought process within special <think>

```

"This is a math problem.\n"
"Please reason step by step, and put your final
answer within \boxed{ }.\n\n"
f"Question: \n{question}\n"

```

Figure 15: Prompt used in the main quantitative reasoning experiments.

```

System Prompt
You are an expert evaluator for mathematical and logical
problems.
Your task is to determine whether a model's answer is correct by
comparing it against the ground truth answer.
Rules:
- Allow for formatting differences (e.g., "0.5" vs "1/2", "x=5" vs
"5", "Option A" vs "A") if the semantic meaning is identical.
- Focus on the final answer, not the reasoning process.
- Output 1 if the answer is correct, 0 if incorrect.
Output format:
Return a single JSON object with exactly the following structure:
{
  "is_correct": 0 | 1,
  "explanation": "Brief explanation of your judgment (one
sentence)"
}
Reply with JSON only. No additional text.
"""

User Prompt
f"""
Question:
{question}

Model's Answer:
{model_answer}

Ground Truth Answer:
{ground_truth}

Please evaluate whether the model's answer is correct and return
the result in JSON format.
"""

```

Figure 16: System and user prompts used for the automatic judge.

and `</think>` tags, and only provide the final answer. It additionally asks the model to keep the reasoning concise rather than producing unnecessarily long chains of thought. We do not use a separate system prompt for this setting.

For the main quantitative reasoning experiments, we use a closely related generation prompt, shown in Figure 15. The prompt again frames the task as a math problem and asks the model to reason step by step, but now requires the final answer to be placed explicitly inside a `\boxed{ }` environment. This formatting constraint simplifies downstream

answer extraction and automatic evaluation. As before, we only use a user prompt and do not specify a system prompt.

To ensure reliable automatic evaluation, we employ a dedicated judging prompt, depicted in Figure 16. The system prompt defines the judge as an expert evaluator for mathematical and logical problems and instructs it to decide whether a model's answer is correct by comparing it with a ground-truth answer. It explicitly allows for benign formatting differences (such as decimals versus fractions or equivalent option labels), and requires the output to be a JSON object containing an `is_correct` flag and a brief textual explanation. In practice, we use this judging prompt with multiple APIs, such as GPT-5-mini and Gemini-2.5-flash, to obtain robust evaluations. When a call fails, we simply retry or fall back to another judge model while keeping the prompt fixed.