Attention-Aligned Reasoning for Large Language Models

Hongxiang Zhang and Yuan Tian and Tianyi Zhang
Purdue University
hxxzhang@purdue.edu
tian211@purdue.edu
tianyi@purdue.edu

Abstract

Large Language Models (LLMs) tend to generate a long reasoning chain when solving complex tasks. However, as the reasoning chain extends, critical intermediate steps and the original prompt will be buried in the context, receiving insufficient attention and leading to errors. In this work, we present ATAR, a novel reasoning method that leverages the inherent reasoning structure to steer LLM attention. Our experiments show that ATAR outperforms SOTA methods across six benchmarks, achieving up to 15.39% absolute improvement. Furthermore, with ATAR, "non-reasoning" models achieve comparable or even better performance compared to reasoning models of the same size in most benchmarks. Finally, our ablation studies show that the attention alignment component contributes significantly, and that these improvements are persist under different attention-steering backends.

1 Introduction

Solving complex tasks requires sophisticated reasoning capabilities. In practice, LLMs tend to generate long reasoning chains. However, as the context grows, the original prompt and key intermediate conclusions can be buried in the middle, receive insufficient attention, and consequently induce errors (Liu et al., 2024). This degradation has been evidenced in many studies in the past (Gu et al., 2025; Chi et al., 2024; Liu et al., 2024; Sun et al., 2024; Yao et al., 2021; Tian and Zhang, 2025; Li et al., 2024; Hong et al., 2025). For example, relevant information placed in the middle of the prompt is often ignored and shows limited influence on generation (Liu et al., 2024). More recently, researchers have identified an interesting phenomenon called "attention sink" (Xiao et al., 2024), where LLMs often allocate disproportionate attention to initial tokens. Excessive fixation on the static preamble can prevent the model from adequately attending to dynamic, task-specific constraints that emerge later in the reasoning process. Together, these effects may make models drift from the original intent and overlook critical intermediate constraints as the context grows.

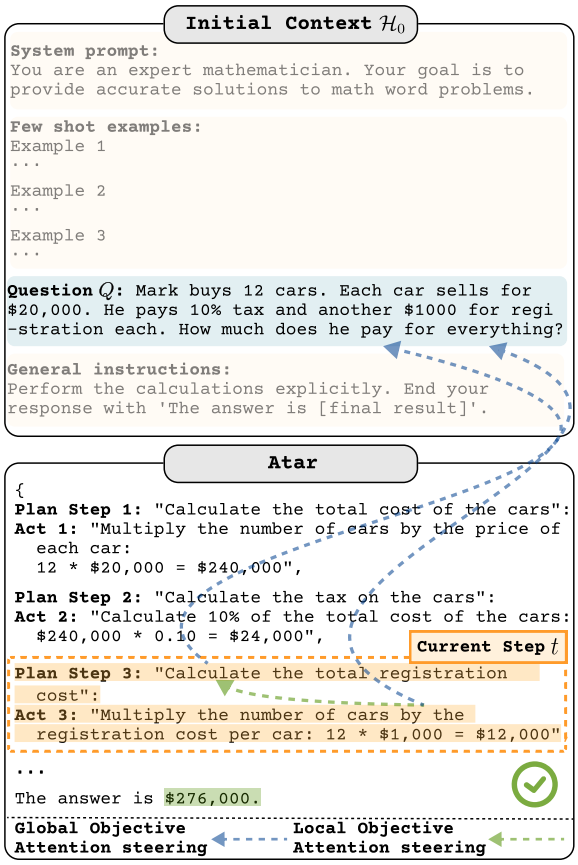


Figure 1: Overview of ATAR. By interleaving planning and action steps, our method explicitly steers attention to balance the global objective (the original question) and the local objective (the current plan step).

Although attention steering methods (Tian and Zhang, 2025; Zhang et al., 2024a) have shown promise in steering attention toward salient content, they introduce their own limitations for multi-step reasoning. First, existing methods such as SPA (Tian and Zhang, 2025) and PASTA (Zhang et al., 2024a) rely on humans to select important

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 25046–25064
October 24-29, 2026 ©2026 Association for Computational Linguistics

tokens to pay attention to, which fails to generalize because the most relevant context shifts dynamically at each reasoning step. Second, these methods apply a fixed steering strength during generation. Such a static intervention can either be too weak to prevent drift or too strong to preserve fluent, correct reasoning.

To address these limitations, we propose ATAR (ATtention-Aligned Reasoning), a reasoning method that explicitly balances attention between the overarching goal and intermediate reasoning steps by leveraging the inherent structure of the reasoning chain. Figure 1 illustrates our method with an example where the current step provides the immediate sub-goal and the original user prompt specifies the global objective. As sub-goals evolve over the reasoning chain, the required degree of attention steering may vary at different steps. To accommodate this, we introduce a dynamic, step-level mechanism that adjusts the steering strength based on model confidence at each reasoning step.

We evaluate ATAR on six benchmarks and six base LLMs with varying sizes. Compared with SOTA reasoning methods such as Tree-of-Thought (Yao et al., 2023) and Re-Reading (Xu et al., 2024), ATAR consistently achieves better performance by up to 15.39% absolute improvement. Furthermore, compared with reasoning models such as Qwen3-thinking (Team, 2025) and Phi-4-reasoning (Abdin et al., 2025), ATAR helps the corresponding non-reasoning models in the same model family achieve better or comparable performance in most settings. Our ablation studies show that the attention alignment component contributes significantly, with improvements persisting across different attention-steering backends.

2 Method

Attention Aligned Reasoning. In complex reasoning tasks, the input context typically includes diverse information components, such as the system prompt, few-shot examples, relevant knowledge, and general instructions. Within this context, the core question Q, is often embedded in the middle. For simplicity, we denote Q as the original question and the contextual information excluding Q as the preamble variable Φ. The initial context is thus defined as $\mathcal{H}_0 = \{\Phi, Q\}$. In our method, we posit Q as the global objective that should consistently guide the generation process.

Prior research has demonstrated the effectiveness of the "Plan-then-Solve" paradigm for complex reasoning, in which a model generates a comprehensive plan before executing step-by-step actions (Zhou et al., 2023a; Jiang et al., 2024; Zhou et al., 2024; Wang et al., 2023). However, when executing later actions, the corresponding plan steps may be separated by a large amount of intervening context, interfering the generation. As illustrated in Figure 2, as the model generates actions, the corresponding plan step may be obscured by the accumulation of intermediate context, leading to attention drift. Inspired by findings that LLMs often attend to immediately generated tokens (Liu et al., 2024), we place each action immediately following the corresponding plan step. Thus, we formalize the reasoning process as a trajectory τ comprising T steps, where each step t interleaves a planning step s_t and an action step a_t. The complete trajectory is represented as $\tau = \{s_1, a_1, \ldots, s_T, a_T\}$.

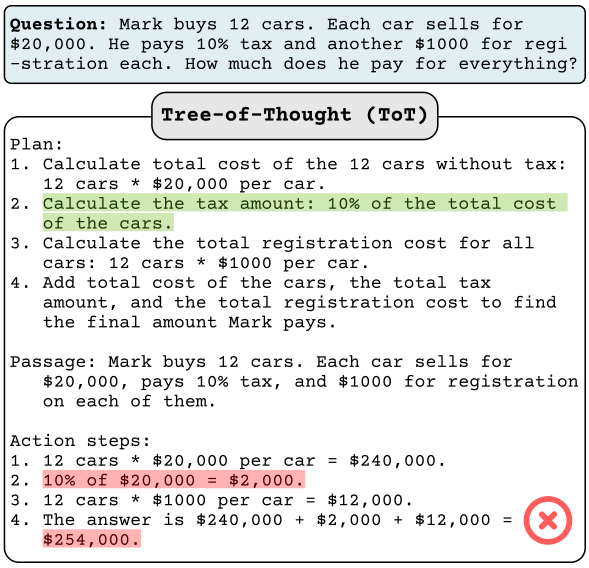


Figure 2: An example in Tree-of-Thought. ToT successfully generates a correct plan. The corresponding action is separated from the plan by a large amount of intervening context, thus deviating from the plan's instruction.

Within this structure, we found that plans offer distinct reference points that can naturally serve to guide attention alignment. Each plan step s_t explicitly specifies the sub-goal to be accomplished at step t. Therefore, we interpret the current plan step s_t as the *local objective*, while the original query Q defines the *global objective*. To generate the action a_t, ATAR steers the model's attention to both the global objective Q and the local objective s_t. This allows the model to balance attention between the overarching goal and intermediate reasoning steps:

$$a_t = f(\mathcal{H}_0 \oplus \tau_{t-1} \oplus s_t, \{Q, s_t\}). \quad (1)$$

where we denote the generation function with attention steering as $f(x, \mathcal{S})$, where x represents the sequence of preceding tokens (i.e., the context) and $\mathcal{S}$ represents the specific set of tokens to which the model's attention is steered. The $\oplus$ indicates string concatenation.

Similarly, generating the next plan step s_t may be interfered by the growing context, particularly from previously generated actions. To ensure that new sub-goals remain faithful to the original question, ATAR steers the generation of each plan step toward the global objective Q:

$$s_t = f(\mathcal{H}_0 \oplus \tau_{t-1}, \{Q\}). \tag{2}$$

Dynamic Attention Alignment. As reasoning trajectories progress, the required degree of attention alignment may vary at different steps. Prior work has explored using an LLM's predicted probability distribution as an approximated signal to assess prediction quality (Geng et al., 2024; Fu et al., 2026; Jiang et al., 2023; Spiess et al., 2025; Kadavath et al., 2022; Orgad et al., 2025). High token-level confidence generally corresponds to locally stable reasoning (Varshney et al., 2022; Jiang et al., 2021), where the model's internal trajectory is already coherent and requires only weak steering. Low confidence, on the other hand, is often correlated with uncertainty or inconsistency. In such cases, stronger steering helps realign the model toward the relevant context.[1]

Therefore, we introduce a step-level steering strength ω_j that is dynamically adjusted based on model confidence.[2] Let $P_j = \{p_{j,1}, p_{j,2}, \ldots, p_{j,m}\}$ denote the probabilities of the m tokens generated at step j. We define the step confidence as the harmonic mean of probabilities:

$$\text{conf}(P_j) = \frac{m}{\sum_{k=1}^{m} \frac{1}{p_{j,k}}} \tag{3}$$

We adopt the harmonic mean because it is sensitive to low-probability tokens (Kai et al., 2024; Krishnan et al., 2024), so a few uncertain predictions can sharply lower the overall score. This behavior is well-suited to multi-step reasoning, where attention drift is often associated with locally uncertain generations (Kai et al., 2024). By comparison, alternatives such as entropy or the arithmetic mean smooth uncertainty more evenly across tokens and thus may fail to emphasize the low-confidence tail, where intervention is most critical. We experimented with alternative designs in Appendix A.2 and demonstrated that harmonic mean achieves the best performance.

We then calculate the dynamic strength ω_j inversely with this confidence:

$$\omega_j = \frac{\omega \cdot \sum_{k=1}^{m} \frac{1}{p_{j,k}}}{m}, \tag{4}$$

where ω is the base steering strength and ω_j[3] is the steering strength at decoding step j. Consequently, attention steering is amplified during highly uncertain decoding steps and attenuated when the model is confident.

To implement the actual attention intervention, we adopt Selective Prompt Anchoring (SPA) (Tian and Zhang, 2025) as a lightweight steering backend. We experimented with an alternative attention steering method, PASTA (Zhang et al., 2024a), and demonstrated that our method achieves persistent performance with the alternative steering method in Sec. 3.4.

3 Experiments

3.1 Benchmarks

We evaluated ATAR on six benchmarks. The first three benchmarks are about math reasoning, including GSM8K (Cobbe et al., 2021), AQuA (Ling et al., 2017), and MATH-500 (Lightman et al., 2024). These three benchmarks span increasing difficulty levels, from grade-school math problems (Cobbe et al., 2021) to GMET/GRE (Ling et al., 2017) level and competition-level problems (Hendrycks et al., 2021). The next two benchmarks are about commonsense reasoning, including StrategyQA (Geva et al., 2021) and Thinking for Doing (T4D) (Zhou et al., 2023b). Finally, following Zou et al. (2025); Juneja et al. (2025); Wei et al. (2022), we select challenging tasks in BIG-Bench Hard (BBH) (Suzgun et al., 2023) as the last benchmark. Specifically, this benchmark covers six types of reasoning tasks, including *Disambiguation QA* (DQA), *Causal Judgment* (CJ), *Date Understanding* (DU), *Logical Deduction* (LD), *Salient Translation Error*

[1] While token-level confidence is not universally reliable and can be mis-calibrated, we adopt it as a lightweight, approximate indicator of local stability.

[2] Our strength adjustment strategy builds upon the confidence-modulated strength strategy in SPA (Tian and Zhang, 2025). While SPA adjusts the strength based on confidence for each logit at the vocabulary level, we introduce an additional factor to adjust the strength for each step.

[3] We clip ω_j to the range recommended by SPA (Tian and Zhang, 2025). Small probabilities can increase steering, as intended, but cannot produce unbounded numerical values.

Detection (STED), and *Snarks*. We report final-answer accuracy across all benchmarks.[4]

3.2 Models and Baselines

Base Models. We experiment with six non-reasoning LLMs spanning various scales as the backbone of ATAR. We select Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Llama-3.2-3B-Instruct (Grattafiori et al., 2024), Phi-4-mini (Abouelenin et al., 2025), Qwen3-4B-Instruct-2507 (Team, 2025), Phi-4 (Abdin et al., 2024), and Qwen3-30B-A3B-Instruct-2507 (Team, 2025).

Comparison Baselines. We compare our method against four representative reasoning methods for LLM reasoning. First, we include **Chain-of-Thought (CoT)** (Wei et al., 2022; Kojima et al., 2022), a widely used baseline for LLM reasoning. Second, we include **Plan-and-Solve+ (PS+)** (Wang et al., 2023), which first generates a plan and then solves the problem. Third, we include **Re-Reading (RE2)** (Xu et al., 2024), which asks the model to read the question again and then solve the problem. Lastly, we include **Tree-of-Thought (ToT)** (Yao et al., 2023), which explores multiple reasoning paths through deliberate tree search.

Furthermore, we consider four reasoning LLMs that have a comparable size to the non-reasoning base models used in ATAR as baselines to investigate whether non-reasoning LLMs combined with ATAR achieve competitive performance against reasoning models across different model scales, including Phi-4-mini-reasoning (Abouelenin et al., 2025), Qwen3-4B-Thinking-2507 (Team, 2025), Phi-4-reasoning (Abdin et al., 2024), and Qwen3-30B-A3B-Thinking-2507 (Team, 2025).

3.3 Main Results

Math Reasoning. Math reasoning represents one of the most challenging aspects of LLM reasoning capabilities. As shown in Table 1, ATAR consistently improves significant accuracy across three math benchmarks. Specifically, it achieves absolute improvements of around 5% on GSM8K, AQuA and up to 8.4% on MATH-500 across most models.

Although PS+, RE2, and ToT demonstrate some effectiveness in improving LLM's math reasoning capabilities, Table 1 shows that such improvement is not consistent. For some benchmarks and base models, these methods achieve even worse performance compared with CoT. Specifically, they tend to be more effective on larger LLMs (Phi-4-15B and Qwen3-30B). This may be because larger models are more capable of following instructions and external guidance to align reasoning trajectories.

As mentioned in Section 3.1, GSM8K, AQuA, and MATH-500 represent three levels of math reasoning difficulty. Table 1 shows that ATAR consistently improves the performance of different LLMs across all three benchmarks, while the other three methods show different levels of performance degradation compared to CoT on certain models and benchmarks. This highlights the robustness and generalization of ATAR across diverse model architectures and reasoning difficulty levels.

Commonsense Reasoning. StrategyQA requires multi-hop reasoning over commonsense knowledge. As shown in Table 1, ATAR persistently improves accuracy across six base LLMs. In contrast, PS+, RE2, and ToT occasionally outperform the baseline CoT method.

T4D is a grounded Theory of Mind reasoning task (Baron-Cohen et al., 1985; Frith and Frith, 2003), which requires the model to predict reasonable actions based on a character's hidden thoughts and perspective. PS+, RE2, and ToT exhibit mixed performance on T4D. They tend to be more effective in larger models. In comparison, ATAR demonstrates significant performance gains of over 9% on four of the six LLMs.

BBH. BBH aggregates challenging algorithmic and symbolic tasks. ATAR demonstrates average performance gains ranging from 1.61% to 15.39%[5]. When analyzing specific task categories in Table 2, we find that those requiring tracking of intermediate reasoning benefit the most, for example *date understanding*, and *logical deduction*. We attribute this to ATAR's attention steering that augments critical reasoning steps and the original question throughout generation. In contrast, PS+, RE2, and ToT show inconsistent improvements.

Comparison to RL-enhanced Reasoning Models Recent advances in reasoning capabilities have been dominated by reinforcement learning-enhanced "reasoning" models that employ extensive internal reasoning chains during inference. However, these models are costly to fine-tune and require large-scale training data. This raises a question: *Can non-reasoning LLMs combined with*

[4]Prompt templates and evaluation details are provided in Appendix E.

[5]We detail the subtask performance in Appendix G

Table 1: Evaluation results on six benchmarks. Best results are shown in **bold**, and those indicating a performance gain compared to Chain-of-Thought are shown in green.

Model	Method	Math			CommonSense		BBH
		GSM8K	AQuA	MATH-500	StrQA	T4D	
Llama3.2-3B	CoT	73.16	46.06	40.80	66.64	32.98	35.09
	PS+	76.95 (+3.79)	38.58 (-7.48)	42.20 (+1.40)	62.79 (-3.85)	31.81 (-1.17)	40.51 (+5.42)
	RE2	74.68 (+1.52)	47.28 (+1.22)	41.00 (+0.20)	65.03 (-1.61)	30.32 (-2.66)	38.02 (+2.93)
	ToT	70.89 (-2.27)	35.04 (-11.02)	32.80 (-8.00)	60.44 (-6.20)	40.80 (+7.82)	41.32 (+6.23)
	ATAR	**78.32 (+5.16)**	**48.43 (+2.37)**	**42.40 (+1.60)**	**67.55 (+0.91)**	**43.79 (+10.81)**	**50.48 (+15.39)**
Phi-4-mini	CoT	84.53	61.81	61.20	67.03	39.54	60.51
	PS+	88.01 (+3.39)	62.02 (+0.21)	62.40 (+1.20)	58.86 (-8.17)	41.16 (+1.62)	59.63 (-0.88)
	RE2	87.33 (+2.80)	59.06 (-2.75)	60.80 (-0.40)	61.83 (-5.20)	44.50 (+4.96)	61.39 (+0.88)
	ToT	85.22 (+0.69)	66.93 (+5.12)	57.60 (-3.60)	61.22 (-5.81)	48.05 (+8.51)	51.72 (-8.79)
	ATAR	**88.02 (+3.49)**	**68.50 (+6.69)**	**62.60 (+1.40)**	**68.69 (+1.66)**	**49.47 (+9.93)**	**62.42 (+1.91)**
Qwen3-4B	CoT	92.27	73.62	80.20	68.03	70.21	73.33
	PS+	91.43 (-0.84)	71.65 (-1.97)	82.40 (+2.20)	68.56 (+0.53)	59.82 (-10.39)	70.99 (-2.34)
	RE2	88.70 (-3.57)	75.98 (+2.36)	81.00 (+0.80)	69.83 (+1.80)	67.02 (-3.19)	**75.75 (+2.42)**
	ToT	89.79 (-2.48)	74.80 (+1.18)	86.60 (+6.40)	69.89 (+1.86)	52.24 (-17.97)	69.82 (-3.51)
	ATAR	**93.56 (+1.29)**	**79.92 (+6.30)**	**88.60 (+8.40)**	**70.13 (+2.10)**	**71.56 (+1.35)**	75.31 (+1.98)
Llama3.1-8B	CoT	77.03	50.79	43.40	70.24	26.77	49.45
	PS+	77.18 (+0.15)	48.65 (-2.14)	40.60 (-2.80)	65.85 (-4.39)	28.79 (+2.02)	51.72 (+2.27)
	RE2	72.10 (-4.93)	51.97 (+1.18)	46.20 (+2.80)	69.74 (-0.50)	28.55 (+1.78)	54.58 (+5.13)
	ToT	70.28 (-6.75)	52.76 (+1.97)	43.60 (+0.20)	69.58 (-0.66)	35.75 (+8.98)	50.47 (+1.02)
	ATAR	**83.93 (+6.90)**	**55.51 (+4.72)**	**50.20 (+6.80)**	**73.54 (+3.30)**	**40.01 (+13.24)**	**58.53 (+9.08)**
Phi-4	CoT	89.08	68.11	74.60	77.51	73.94	72.08
	PS+	91.36 (+2.28)	72.75 (+4.64)	76.40 (+2.00)	75.59 (-1.92)	76.24 (+2.30)	68.42 (-3.66)
	RE2	89.84 (+0.76)	69.29 (+1.18)	74.80 (+0.20)	76.90 (-0.61)	75.79 (+1.85)	74.94 (+2.86)
	ToT	90.60 (+1.52)	75.67 (+7.56)	79.40 (+4.80)	70.02 (-7.49)	83.95 (+10.01)	64.10 (-8.70)
	ATAR	**94.38 (+5.30)**	**79.13 (+11.02)**	**81.00 (+6.40)**	**77.82 (+0.31)**	**85.99 (+12.05)**	**75.31 (+3.23)**
Qwen3-30B	CoT	88.07	81.10	89.40	78.60	84.92	72.80
	PS+	89.04 (+0.97)	80.83 (-0.27)	91.20 (+1.80)	78.70 (+0.10)	82.62 (-2.30)	71.79 (-1.01)
	RE2	90.54 (+2.47)	82.28 (+1.18)	89.80 (+0.20)	78.91 (+0.31)	**89.36 (+4.44)**	**75.68 (+2.88)**
	ToT	90.43 (+2.36)	**87.06 (+5.96)**	91.60 (+2.20)	77.41 (-1.19)	83.97 (-0.95)	73.55 (+0.75)
	ATAR	**92.16(+4.09)**	83.46 (+2.36)	**93.60 (+4.20)**	**79.65 (+1.05)**	85.56 (+0.64)	74.43 (+1.63)

Table 2: Performance of ATAR across BBH task categories. Full task names are provided in Sec. 3.1.

Model	BBH					
	DQA	CJ	DU	LD	STED	Snarks
Llama3.2-3B	52.10	54.01	52.40	40.40	38.40	74.47
Phi4-mini	60.80	62.03	69.60	51.20	56.00	79.78
Qwen3-4B	74.00	61.50	77.20	89.60	64.40	84.27
Llama3.1-8B	60.80	53.48	64.00	51.20	52.80	71.35
Phi-4	70.40	65.24	87.20	81.20	59.20	90.45
Qwen3-30B	72.00	65.78	78.40	80.00	64.00	88.20

ATAR *achieve competitive performance against reasoning models?*

To investigate this question, we compare our method applied to non-reasoning LLMs against the corresponding reasoning models version. As illustrated in Table 3, applying ATAR to non-reasoning models achieves competitive or superior performance compared to RL-enhanced reasoning models on most benchmarks and models. Specifically, ATAR closes the performance gap significantly in three math benchmarks; On commonsense reasoning tasks and BBH, ATAR exceeds reasoning model performance on most benchmarks and LLMs.

Notably, reasoning models show advantages on tasks where the non-reasoning models have weak baseline performance. Specifically, Phi-4, Qwen3-4B, and Phi-4-mini show large gaps on MATH-500. We hypothesize that this pattern is largely attributable to reinforcement learning in post-training. RL is particularly effective for verifiable reasoning tasks such as mathematics, where correctness can be directly evaluated and used as a reliable learning signal (Team, 2025; Guo et al., 2025). In contrast, for tasks where non-reasoning models already demonstrate strong performance, reinforcement learning provides limited improvement. This pattern is also observed in Kirk et al. (2024). Nevertheless, ATAR shows consistent performance improvements across all tasks and difficulty levels.

Table 3: Evaluation comparison with reasoning models within the same family and size.

Model	Math			CommonSense		BBH
	GSM8K	AQuA	MATH-500	StrQA	T4D	
Phi-4-mini-reasoning	85.37	67.97	**87.80**	66.38	45.04	59.85
Phi-4-mini (w/ **ATAR**)	**88.02**	**68.50**	62.60	**68.69**	**49.47**	**62.42**
Qwen3-4B-Thinking	92.27	67.32	**93.20**	68.31	**73.40**	**75.34**
Qwen3-4B (w/ **ATAR**)	**93.56**	**79.92**	88.60	**70.13**	71.56	75.31
Phi-4-reasoning	91.20	**83.20**	**91.20**	77.61	74.11	74.98
Phi-4 (w/ **ATAR**)	**94.38**	79.13	81.00	**77.82**	**85.99**	**75.31**
Qwen3-30B-A3B-Thinking	**93.75**	83.26	**95.60**	77.26	80.96	**76.54**
Qwen3-30B (w/ **ATAR**)	92.16	**83.46**	93.60	**79.65**	**85.56**	74.43

Table 4: Performance of ATAR under different backends, including SPA (Tian and Zhang, 2025), and PASTA (Zhang et al., 2024a).

Method	Math	CommonSense	BBH
CoT	24.17	57.59	31.67
PS+	23.66	55.02	32.89
RE2	24.96	58.38	33.60
ToT	22.56	49.46	28.74
ATAR (w/ SPA)	**29.17**	59.77	**35.01**
ATAR (w/ PASTA)	26.53	**60.43**	33.71

In summary, instead of learning implicit reasoning patterns through expensive training, ATAR leverages the explicit reasoning structure for attention alignment to improve the reasoning capabilities. These findings suggest that ATAR can serve as an effective alternative to computationally expensive RL-enhanced reasoning.

3.4 Generalizability to Attention Steering Backbones

ATAR is orthogonal to the choice of attention steering method. To validate this, we experiment ATAR with an alternative steering backend, PASTA (Zhang et al., 2024a). In this experiment, we follow the original PASTA setup with Llama (Touvron et al., 2023), since PASTA only supports limited backbones and defaults to Llama for evaluation. Accordingly, we evaluate ATAR with PASTA across three representative benchmarks from our six primary evaluation benchmarks: AQuA, StrQA, and BBH.

As shown in Table 4, ATAR consistently improves performance over other reasoning methods under different attention steering backends. These results suggest that ATAR generalizes across attention steering backbones.

3.5 Ablation Study

Attention Steering. To isolate the contribution of the attention steering, we compare ATAR against a variant without attention steering. Additional steering strategies and case studies are provided in Appendix A and Appendix F.

As shown in Table 5, ATAR consistently outperformed its variant without attention steering across all benchmarks. This ablation confirms that planning and structured reasoning alone are insufficient to prevent attention drift.

Dynamic Attention Steering. We further compare ATAR against an ablated version that only performs static attention steering without dynamic token selection and strength tuning. Following the main experiments, we still use SPA (Tian and Zhang, 2025) as the attention steering backbone. Note that without dynamic token selection, we need to manually select which tokens to steer attention to during the reasoning and generation process, which does not scale to a large experiment. Therefore, we follow the original setup from SPA, which steers to the original user question for simplicity and automation.

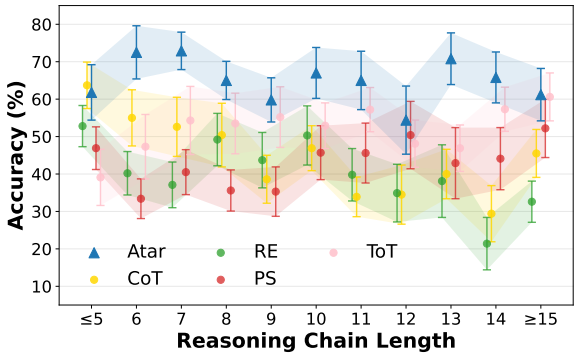
As shown in Table 5, ATAR consistently outperformed its variant without the dynamic mechanism across all benchmarks. These results indicate that simply combining planning with attention steering is insufficient. The gains of ATAR stem from dynamic attention steering, which maintains focus as reasoning evolves.

3.6 Reasoning Length and Task Complexity

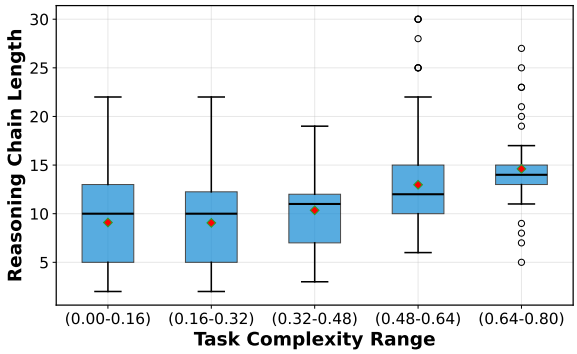
To understand the robustness of ATAR to reasoning length and task difficulty, we conducted more detailed analyses. For each benchmark, we randomly sample 200 questions for analysis. Following prior work (Wu et al., 2026; Jin et al., 2024), we quantify

Table 5: Ablation study comparing ATAR with and without attention steering. Results show that attention steering consistently improves performance. Best results are highlighted in **bold**.

Model	Method	Math			CommonSense		BBH
		GSM8K	AQuA	MATH-500	StrQA	T4D	
Llama3.1-8B	ATAR (w/o attention steering)	79.01	53.15	47.60	71.17	35.28	52.77
	ATAR (w/o dynamic mechanism)	80.13	48.03	48.80	73.18	38.17	53.64
	ATAR	**83.93**	**55.51**	**50.20**	**73.54**	**40.01**	**58.53**
Phi-4-mini-4B	ATAR (w/o attention steering)	78.77	58.66	55.40	68.60	38.83	57.44
	ATAR (w/o dynamic mechanism)	84.32	62.60	57.20	67.97	48.47	60.80
	ATAR	**88.02**	**68.50**	**62.60**	**68.69**	**49.47**	**62.42**



(a) Reasoning chain length and accuracy comparison with 95% confidence intervals.



(b) Tasks Complexity and reasoning chain length comparison. The lower complexity values indicate easier tasks.

Figure 3: Analysis of chain length and task complexity.

reasoning length by segmenting each model output into discrete reasoning steps and using the number of steps as a measure of reasoning depth. Task complexity is defined as $1 -$ accuracy, where higher values indicate more challenging questions[6].

Robustness to Reasoning length. We first examine the robustness of ATAR with respect to varying reasoning lengths. Figure 3a illustrates the accuracy of ATAR compared to other baselines, across varying reasoning lengths. ATAR achieves superior accuracy over the baselines across most reasoning lengths. While it consistently surpasses other methods, there is a slight performance dip relative to CoT on short sequences (length ≤ 5). This can be attributed to the design of ATAR, which is intended to mitigate attention drift in longer multi-step reasoning. Since simple tasks often can be solved in a few steps, they suffer less from attention drift. Moreover, ATAR exhibits stable and high accuracy even as the reasoning chain expands. These results demonstrate that ATAR is robust to reasoning length, effectively handling tasks regardless of whether they require short or extended reasoning chains.

Scaling Reasoning Length with Complexity. Next, we analyze how reasoning chain length scales with task difficulty. Figure 3b shows the distribution of successful reasoning lengths across tasks of varying difficulty. The results show a clear trend that as complexity increases, ATAR tends to generate longer reasoning chains. To quantify this trend, we apply the Jonckheere-Terpstra test and obtain $\tau = 0.102$ with $p = 1.279 \times 10^{-7}$, indicating a statistically significant increasing trend across complexity levels. The effect size is modest, which is expected given variability in solution strategies and the large sample size. This aligns with the observation in Wu et al. (2026) that harder problems require longer reasoning chains to solve. We attribute this capability to the attention steering mechanism, which enables the model to focus on both problem context and immediate reasoning object throughout the reasoning, preventing attention drift as the reasoning chain extends.

In summary, our analysis highlights two takeaways: (1) ATAR demonstrates robustness across varying reasoning lengths. (2) ATAR encourage the model to generate longer reasoning chains for difficult problems, supporting its effectiveness in scaling to complex tasks.

[6]Details on reasoning step segmentation and task complexity computation are provided in Appendix E.

3.7 Failure Case Analysis

To understand the failure modes in ATAR. We conducted a manual failure analysis on 300 randomly sampled failed cases (50 from each benchmark). Our analysis identified four primary failure modes:

Reasoning Errors (42%). The most frequent failure mode involves the LLM making mistakes during the reasoning process. These include the misapplication of causal principles, flawed deductions, and incorrect conditional reasoning. For example, models sometimes exhibit causal misjudgment, where a condition that is necessary but not sufficient is incorrectly treated as the sole cause of an outcome. We provide a concrete instance of this phenomenon from the BBH benchmark in Appendix F.1. Such errors suggest that, while attention steering helps maintain focus on relevant steps, it cannot fully compensate for weak logical priors or gaps in world knowledge.

Misunderstanding the Problem (30%). Another common failure mode stems from misunderstanding complex questions, leading to errors such as misidentifying all variables, misinterpreting the problem's requirements, or incorrectly parsing the relationships between entities. This issue is most common in multi-variable mathematical reasoning tasks, such as AQua. We provide a representative example in Appendix F.2, where the model assigns wrong numerical values to the entity, causing the subsequent reasoning to an incorrect answer.

Computational Errors (21.5%). These failures include arithmetic mistakes, unit conversion errors, or algebraic slips. Even when the reasoning chain is correct, a single miscalculation often propagates to the final incorrect answer.

Sensitivity to Plan Quality (6.5%). As plan quality might affect downstream reasoning, we specifically examined the impact of planning quality within this category. We observed that only 6.5% of total failures were attributable to planning errors. To determine if ATAR amplifies these errors, we cross-referenced these cases with one of the strongest baselines (RE2). We found that only 2.3% of these specific cases were solvable by RE2. This suggests that when plans fail, the root cause is typically the intrinsic difficulty of the problem, not the anchoring mechanism. In other words, ATAR does not amplify errors. It reflects the model's existing understanding. Incorrect plans are strongly correlated with questions the base model already struggles with, regardless of attention steering.

4 Related Work

LLM Reasoning. Prompt engineering has been widely adopted as a fundamental approach for enhancing LLM reasoning capabilities (Liu et al., 2023; Brown et al., 2020). Pioneered by Chain-of-Thought prompting (Wei et al., 2022), encourages explicit intermediate steps, which significantly improve performance on multi-step reasoning tasks. This has inspired numerous derivatives, from problem decomposition methods (Wang et al., 2023; Zhou et al., 2023a; Khot et al., 2023; Drozdov et al., 2022), to enhancing query comprehension (Xu et al., 2024; Zheng et al., 2024; Mekala et al., 2024; Deng et al., 2023; Mishra and Nouri, 2023). While these methods demonstrate effectiveness in specific domains, they rely on predefined, static prompt for different tasks. On the other hand, LLMs remain sensitive to prompt variations and suffer from attention dilution during long generations (Liu et al., 2024; Li et al., 2024; Hong et al., 2025; Lu et al., 2022; Gu et al., 2025). ATAR addresses these limitations by enabling the model to recalibrate its focus to the global objective defined by the original question and the local objective established by the current plan step at each stage of reasoning.

Attention steering. In contrast to the aforementioned prompt engineering, which devises better prompt strategies, attention steering methods directly guide LLMs during inference to emphasize the user-specified part of the context. Specifically, SPA (Tian and Zhang, 2025) adjusts the logit probability distribution to emphasize the specified context. PASTA (Zhang et al., 2024a) identifies and reweights a subset of attention heads to redirect the model's attention to user-specified parts. SSA (Zhang et al., 2024b) augments softmax with temperature scaling, whereas TOAST (Shi et al., 2023) guides attention using learned feature selection. However, these methods are static and task-specific, limiting their adaptability to diverse reasoning contexts. ATAR addresses this limitation by leveraging structured intermediate representations to enable dynamic attention steering.

5 Conclusion

We presented ATAR, a reasoning method that leverages the inherent structure of the reasoning chain to dynamically align attention during reasoning. Across six diverse reasoning benchmarks, ATAR consistently outperforms existing baselines. Notably, ATAR enhanced "non-reasoning" models

achieve competitive performance compared with specialized reasoning models under most benchmarks. Moreover, our analysis reveals that ATAR's advantages are robust to varying task complexities. We hope ATAR serves as a step toward more reliable LLMs reasoning that requires neither parameter updates nor additional sampling.

Limitations

Our method leverages the inherent structure of the reasoning chain to dynamically align attention during generation. However, as detailed in subsection 3.7, ATAR does not address intrinsic error modes of large language models. In particular, it cannot correct fundamental misunderstandings of the problem, nor can it reliably prevent arithmetic mistakes or algebraic slips. When such errors occur early, they may still propagate through subsequent steps despite improved attention alignment.

Furthermore, the headroom for inference-time intervention naturally diminishes when applying ATAR to larger or more powerful models, which suffer less from attention misalignment compared to smaller models. Nevertheless, ATAR still demonstrates performance gains on Qwen3-30B across all evaluated benchmarks, demonstrating its efficacy even at larger model scales.

Acknowledgments

We thank the anonymous reviewers for their insightful feedback and the time and effort they devoted to reviewing our work. This work was supported in part by NSF grant CCF-2340408.

References

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, and 1 others. 2025. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, and 1 others. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*.

Simon Baron-Cohen, Alan M Leslie, and Uta Frith. 1985. Does the autistic child have a "theory of mind"? *Cognition*, 21(1):37–46.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ta-Chung Chi, Ting-Han Fan, and Alexander Rudnicky. 2024. Attention alignment and flexible positional embeddings improve transformer length extrapolation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 132–148, Mexico City, Mexico. Association for Computational Linguistics.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Yihe Deng, Weitong Zhang, Zixiang Chen, and Quanquan Gu. 2023. Rephrase and respond: Let large language models ask better questions for themselves. *arXiv preprint arXiv:2311.04205*.

Andrew Drozdov, Nathanael Schärli, Ekin Akyürek, Nathan Scales, Xinying Song, Xinyun Chen, Olivier Bousquet, and Denny Zhou. 2022. Compositional semantic parsing with large language models. *arXiv preprint arXiv:2209.15003*.

Uta Frith and Christopher D Frith. 2003. Development and neurophysiology of mentalizing. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1431):459–473.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2026. Deep think with confidence. In *International Conference on Learning Representations (ICLR)*.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595, Mexico City, Mexico. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the*

Association for Computational Linguistics, 9:346–361.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. 2025. When attention sink emerges in language models: An empirical view. In *International Conference on Learning Representations (ICLR)*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

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. *NeurIPS*.

Kelly Hong, Anton Troynikov, and Jeff Huber. 2025. Context rot: How increasing input tokens impacts llm performance. Technical report, Chroma.

Xue Jiang, Yihong Dong, Lecheng Wang, Zheng Fang, Qiwei Shang, Ge Li, Zhi Jin, and Wenpin Jiao. 2024. Self-planning code generation with large language models. *ACM Transactions on Software Engineering and Methodology*, 33(7):1–30.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Mingyu Jin, Qinkai Yu, Dong Shu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. 2024. The impact of reasoning step length on large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1830–1842.

Gurusha Juneja, Gautam Jajoo, Hua Li, Jian Jiao, Nagarajan Natarajan, and Amit Sharma. 2025. Task facet learning: A structured approach to prompt optimization. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 23473–23496.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Jushi Kai, Tianhang Zhang, Hai Hu, and Zhouhan Lin. 2024. SH2: Self-highlighted hesitation helps you decode more truthfully. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4514–4530, Miami, Florida, USA. Association for Computational Linguistics.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks. In *International Conference on Learning Representations (ICLR)*.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of rlhf on llm generalisation and diversity. In *International Conference on Learning Representations (ICLR)*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Ranganath Krishnan, Piyush Khanna, and Omesh Tickoo. 2024. Enhancing trust in large language models with uncertainty-aware fine-tuning. *arXiv preprint arXiv:2412.02904*.

Hanyu Lai, Xiao Liu, Junjie Gao, Jiale Cheng, Zehan Qi, Yifan Xu, Shuntian Yao, Dan Zhang, Jinhua Du, Zhenyu Hou, and 1 others. 2025. A survey of post-training scaling in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2771–2791.

Kenneth Li, Tianle Liu, Naomi Bashkansky, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024. Measuring and controlling instruction (in)stability in language model dialogs. *Preprint*, arXiv:2402.10962.

Zhen Li, Yupeng Su, Songmiao Wang, Runming Yang, Congkai Xie, Aofan Liu, Ming Li, Jiannong Cao, Yuan Xie, Ngai Wong, and 1 others. 2025. Quantization meets reasoning: Exploring and mitigating degradation of low-bit llms in mathematical reasoning. *arXiv preprint arXiv:2505.11574*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word

problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35.

Ruikang Liu, Yuxuan Sun, Manyi Zhang, Haoli Bai, Xianzhi Yu, Tiezheng Yu, Chun Yuan, and Lu Hou. 2025. Quantization hurts reasoning? an empirical study on quantized reasoning models. *arXiv preprint arXiv:2504.04823*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. 2024. Improve mathematical reasoning in language models by automated process supervision. *Preprint*, arXiv:2406.06592.

Anmol Mekala, Anirudh Atmakuru, Yixiao Song, Marzena Karpinska, and Mohit Iyyer. 2025. Does quantization affect models' performance on long-context tasks? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9433–9481.

Raja Sekhar Reddy Mekala, Yasaman Razeghi, and Sameer Singh. 2024. Echoprompt: instructing the model to rephrase queries for improved in-context learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 399–432.

Swaroop Mishra and Elnaz Nouri. 2023. HELP ME THINK: A simple prompting strategy for non-experts to create customized content with models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11834–11890, Toronto, Canada. Association for Computational Linguistics.

OpenAI. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. 2025. Llms know more than they show: On the intrinsic representation of llm hallucinations. In *International Conference on Learning Representations (ICLR)*.

Baifeng Shi, Siyu Gai, Trevor Darrell, and Xin Wang. 2023. Toast: Transfer learning via attention steering. *arXiv preprint arXiv:2305.15542*.

Claudio Spiess, David Gros, Kunal Suresh Pai, Michael Pradel, Md Rafiqul Islam Rabin, Amin Alipour, Susmit Jha, Prem Devanbu, and Toufique Ahmed. 2025. Calibration and correctness of language models for code. In *Proceedings of the IEEE/ACM 47th International Conference on Software Engineering*, ICSE '25, page 540–552. IEEE Press.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. 2024. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and 1 others. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Yuan Tian and Tianyi Zhang. 2025. Selective prompt anchoring for code generation. In *International Conference on Machine Learning*, pages 59528–59551. PMLR.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Neeraj Varshney, Man Luo, and Chitta Baral. 2022. Can open-domain qa reader utilize external knowledge efficiently like humans? *arXiv preprint arXiv:2211.12707*.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.

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.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, and 1 others. 2025. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. 2026. When more is less: Understanding chain-of-thought length in llms. In *International Conference on Learning Representations (ICLR)*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, Jian-Guang Lou, and Shuai Ma. 2024. Re-reading improves reasoning in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15549–15575.

Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. 2021. Self-attention networks can process bounded hierarchical languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3770–3785, Online. Association for Computational Linguistics.

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.

Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. 2024a. Tell your model where to attend: Post-hoc attention steering for LLMs. In *The Twelfth International Conference on Learning Representations*.

Xuechen Zhang, Xiangyu Chang, Mingchen Li, Amit Roy-Chowdhury, Jiasi Chen, and Samet Oymak. 2024b. Selective attention: Enhancing transformer through principled context control. *Advances in Neural Information Processing Systems*, 37:11061–11086.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H Chi, Quoc V Le, and Denny Zhou. 2024. Take a step back: Evoking reasoning via abstraction in large language models. In *International Conference on Learning Representations (ICLR)*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and 1 others. 2023a. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations (ICLR)*.

Pei Zhou, Aman Madaan, Srividya Pranavi Potharaju, Aditya Gupta, Kevin R McKee, Ari Holtzman, Jay Pujara, Xiang Ren, Swaroop Mishra, Aida Nematzadeh, and 1 others. 2023b. How far are large language models from agents with theory-of-mind? *arXiv preprint arXiv:2310.03051*.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V Le, Ed Chi, Denny Zhou, Swaroop Mishra, and Huaixiu Steven Zheng. 2024. Self-discover: Large language models self-compose reasoning structures. *Advances in Neural Information Processing Systems*, 37:126032–126058.

Kaijian Zou, Muhammad Khalifa, and Lu Wang. 2025. On many-shot in-context learning for long-context evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25605–25639.

A Alternative designs

A.1 Attention Steering Strategies

As described in Section 2, our primary ATAR design anchors attention to both the original question and the current plan step during reasoning generation. To validate the design choices of ATAR, we evaluate three alternative anchoring strategies: anchoring solely to the question (*Question-only*), solely to the current plan (*Plan-only*), and to the full plan history (*Plan history*).

We additionally compare against periodic summarization (*Periodic Summarization*), a commonly used context-compression technique in multi-iteration agent trajectories (Wu et al., 2025). Since our setting uses single-shot reasoning (a single, uninterrupted chain of reasoning rather than multi-round interaction), we implemented a periodic summarization baseline where the model is prompted to compress its reasoning history into a concise summary state every K reasoning steps. Results are presented in Table 6.

Plan-only. A core hypothesis of ATAR is that anchoring only to the current sub-goal (plan step) is insufficient, because the model may lose sight of the global objective. Consistent with this intuition, *Plan-only* substantially degrades performance, especially on the weaker Llama-3.1 model. This suggests that step-level guidance alone cannot reliably support long-horizon reasoning.

Impact of History. We also evaluate an alternative design that anchors to all prior plan steps in addition to the current step and the original question. The motivation is that preserving attention to the full reasoning trajectory might provide helpful historical context. Specifically, we compare two anchoring strategies:

- **ATAR (Ours):** Anchors to the original question and current plan step only, where $a_i = \{\text{Question}, \text{Plan}_i\}$
- **Plan history:** Anchors to the original question, current plan step, and all prior plan steps, where $a_i = \{\text{Question}, \text{Plan}_1, \text{Plan}_2, \ldots, \text{Plan}_i\}$

Anchoring to the entire *Plan history* also yields suboptimal results. It is likely because anchoring to all plan history will dilute attention across too many past steps, weakening focus on the immediate reasoning objective.

Periodic Summarization. The result shows that ATAR consistently outperforms the periodic summarization method across all benchmarks and models. This is likely because summary compression introduces information loss and can overwrite fine-grained constraints that remain useful for downstream steps.

In conclusion, our proposed design achieves the best performance. These results support our central design principle: effective reasoning-time attention alignment requires maintaining both global task awareness and step-wise local focus.

A.2 Dynamic Attention Alignment Strength

We evaluate different methods for calculating confidence scores by aggregating token-level probabilities into a step-level confidence score. Let $P_i = \{p_1, p_2, \ldots, p_m\}$ represent the set of token-level confidence scores for tokens generated in the current reasoning step i. We compare three approaches for calculating average confidence scores against a baseline of **Fixed Anchor Strength** (where the dynamic weight is replaced by a constant scalar).

The aggregation methods are defined as follows:

Harmonic Mean:

$$p_{\text{harmonic}} = \frac{n}{\sum_{i=1}^{n} \frac{1}{p_i}} \tag{5}$$

Geometric Mean:

$$p_{\text{geometric}} = \left(\prod_{i=1}^{n} p_i \right)^{1/n} \tag{6}$$

Arithmetic Mean:

$$p_{\text{arithmetic}} = \frac{1}{n} \sum_{i=1}^{n} p_i \tag{7}$$

Lastly, we consider the normalized predictive **Entropy** of the model's next-token distribution.

Table 7 presents the results across two reasoning benchmarks using Llama3.1 and Phi-4-mini models. First, we observe that the Fixed Anchor Strength strategy significantly underperforms compared to dynamic methods, confirming that adaptive anchoring based on model confidence is essential for effective reasoning. Among the dynamic methods, the Harmonic Mean consistently outperforms both geometric and arithmetic means across all settings. This is likely because the harmonic mean is heavily penalized by minimum values. In reasoning tasks, a single low-confidence token signals potential attention drift. The geometric mean

Table 6: Alternative design on anchoring strategies. Results show accuracy (%) on AQuA-RAT and T4D benchmarks.

Model	Method	AQuA-RAT	T4D
Llama3.1-8B	CoT	50.79	26.77
	ATAR (Plan-only)	40.94	26.06
	ATAR (Plan history)	53.15	36.70
	Periodic summarization (K=2)	46.06	39.82
	ATAR (**Ours**)	**55.51**	**40.01**
Phi-4-mini	CoT	61.81	39.54
	ATAR (Plan-only)	50.00	47.34
	ATAR (Plan history)	64.96	47.34
	Periodic summarization (K=2)	57.09	40.88
	ATAR (**Ours**)	**68.50**	**49.47**

Table 7: Alternative design on confidence aggregation methods. “Fixed Strength" uses a constant anchor weight, while others calculate dynamic weights based on the mean token probability. Results show accuracy (%).

Model	Method	AQuA-RAT	T4D
Llama3.1-8B	CoT	50.79	26.77
	ATAR (Fixed Strength)	47.64	36.17
	ATAR (Entropy)	47.64	35.64
	ATAR (Arithmetic)	54.72	35.99
	ATAR (Geometric)	55.11	35.28
	ATAR (Harmonic)	**55.51**	**40.01**
Phi-4-mini	CoT	61.81	39.54
	ATAR (Fixed Strength)	62.60	48.58
	ATAR (Entropy)	60.63	49.30
	ATAR (Arithmetic)	67.71	48.40
	ATAR (Geometric)	67.72	49.11
	ATAR (Harmonic)	**68.50**	**49.47**

performs second-best, as it also penalizes low values more than the arithmetic mean. The arithmetic mean shows the weakest performance, as it can be dominated by high-confidence tokens and may miss instances where attention drift occurs for specific reasoning components. Lastly, the entropy variant does not improve over the harmonic mean and can slightly degrade performance, suggesting that entropy introduces additional noise under our setting and is less aligned with the specific failure patterns that anchoring aims to correct.

B Efficiency

While ATAR outperforms existing methods, we further analyze its efficiency. We use the second per task as the metric. As shown in Table 8, ToT incurs approximately $4\times$ higher latency than ATAR. The high latency of ToT primarily arises from the tree search procedure, which requires generating and evaluating multiple candidate reasoning branches at each step. In contrast, ATAR is modest compared to existing reasoning methods, while delivering substantially higher accuracy. Specifically, to steer attention to specific tokens, it performs an additional masked forward pass to compute logits reweighting.

C Related work: Reasoning LLMs

Recent researches demonstrate specialized training pipelines can internalizing reasoning capabilities directly into the model. State-of-the-art models, such as OpenAI o1 (OpenAI, 2024) and

Table 8: Efficiency comparison

Model	Latency (Second/Task) ↓				
	CoT	PS+	RE2	ToT	ATAR
Llama3.2-3B	34.96	35.20 (+0.24)	27.64 (-7.32)	193.55 (+158.59)	48.06 (+13.10)
Phi-4-mini-4B	26.26	21.58 (-4.68)	24.27 (-1.99)	278.70 (+252.44)	42.30 (+16.04)
Qwen3-4B	61.24	89.97 (+28.73)	50.40 (-10.84)	292.28 (+231.04)	74.29 (+13.05)
Llama3.1-8B	109.82	110.54 (+0.72)	101.31 (-8.51)	428.2 (+318.38)	89.71 (-20.11)
Phi-4-15B	108.44	108.69 (+0.25)	108.66 (+0.22)	482.48 (+374.04)	122.59 (+14.15)
Qwen3-30B	468.99	536.02 (+67.03)	424.9 (-44.09)	1790.36 (+1321.37)	459.03 (-9.96)

DeepSeek-R1 (Guo et al., 2025), leverage fine-tuning and large-scale reinforcement learning (Luo et al., 2024; Lai et al., 2025) on reasoning trajectories, achieving remarkable performance on complex reasoning tasks. Despite their impressive performance, they require substantial computation and a considerable amount of training data. More critically, even powerful reasoning models remain vulnerable to attention dilution during extended generation (Liu et al., 2024; Gu et al., 2025; Xiao et al., 2024), where the model fails to sustain focus on relevant context. In comparison, ATAR provides a training-free reasoning method that uses the inherent structure of the reasoning chain to align attention during generation. Our results show that ATAR enables "non-reasoning" models reach better or comparable performance compared to specialized reasoning models of the same size and family across most settings.

D ATAR Prompt Details

Table 9 presents the prompt used by ATAR to elicit explicit planning and step-by-step reasoning in a structured format. The prompt instructs the model to first decompose the original problem into a sequence of plan steps, followed by corresponding actions and reasoning for each step, ensuring that intermediate reasoning remains aligned with the overall problem specification.

E Implementation and evaluation details

E.1 Prompt example

Chain-of-Thought and Re-Reading. Table 10 presents the prompts used by the Chain-of-Thought and Re-Reading. The Chain-of-Thought prompt encourages step-by-step reasoning, while the Re-Reading prompt reinforces attention to the original question by explicitly repeating it before reasoning.

Plan-and-solve+. We adopt the implementation of Plan-and-Solve+ (Wang et al., 2023). For mathematical reasoning tasks, we use a prompt that explicitly emphasizes variable extraction, planning, and step-by-step numerical reasoning. For non-mathematical (commonsense) tasks, we adopt a simplified planning-based prompt, as shown in Table 11.

E.2 Evaluation Details

Our experiments were conducted on the NVIDIA Quadro RTX 8000 GPU. To improve inference efficiency, we employ quantized inference, specifically NF4 4-bit quantization with FP16 computation. We observe some performance drop compared to full-precision inference, which is consistent with prior studies showing that quantization can negatively affect performance, particularly for smaller models and long reasoning generations (Mekala et al., 2025; Liu et al., 2025; Li et al., 2025).

To verify that our conclusions remain robust under full-precision inference, we further conduct a verification experiment on MATH-500 under full precision. The results show that ATAR consistently outperforms CoT across both models: Llama3.1 improves from 46.8 to 52.2, and Qwen3-4B improves from 81.6 to 90.4. This suggests that while quantization may affect absolute performance, the relative advantage of ATAR remains stable.

During generation, we use greedy decoding. We adopt standard metrics used in prior work (Chuang et al., 2024; Wang et al., 2023; Zhou et al., 2024), including accuracy and exact match, for AQuA,

TASK INSTRUCTIONS

You are an expert problem solver. Your task is to decompose the given problem into a clear, step-by-step plan, reasoning the plan, and solve the problem step by step in JSON format. For each plan step, provide a key-value pair: the key is the plan step (as stated), the value is the detailed reasoning and action for that step. Now, implement a reasoning structure to follow step-by-step and arrive at correct answers in JSON format.

TASK

Original problem: **{question}**

Table 9: Prompt used by ATAR to elicit structured planning and step-wise reasoning in format.

Chain-of-Thought

Let's think step by step.

Re-Reading.

{question}
Read the question again:
{question}

Table 10: Prompt used by Chain-of-Thought and Re-Reading.

BBH, T4D, and MATH. For GSM8K and StrategyQA, we follow the factual accuracy evaluation protocol introduced by Chuang et al. (2024).For the analysis presented in section 3.6, we examine model performance across different reasoning chain lengths on Llama3.2-4B.

To ensure consistent answer extraction, we prompt all models to conclude their response with the phrase: "Conclude with the final answer using the format: "Final answer": "<your answer>" where <your answer> denotes either a multiple-choice option or a string answer. We then apply task-specific heuristics to extract <your answer> from the output.

E.3 Implementation Details

We follow the definitions of reasoning step segmentation and task complexity from (Wu et al., 2026; Jin et al., 2024). The performance gain of a task is computed as the difference in accuracy between two methods on that task.

Reasoning Step Segmentation. To quantify reasoning length, we measure the number of discrete reasoning steps in each model-generated result. Rather than using token count, which can be noisy and model-dependent, we adopt step count as a more meaningful indicator of reasoning depth. Specifically, we segment the full reasoning output by newline characters ("\n"), remove empty lines resulting from consecutive newlines ("\n\n"), and treat each remaining line as one reasoning step. The total number of such lines is taken as the reasoning length for that example.

For datasets with high variability in reasoning length (e.g., MATH-500), we normalize step counts using fixed-width bins to ensure sufficient samples per group. We use a bin width of 5 when comparing reasoning length distributions across models, and a finer bin width of 2 when analyzing the relationship between reasoning length and task complexity. We verify that using smaller bin widths yields qualitatively similar results.

Task Complexity. Task complexity is defined based on empirical difficulty. For each question, we compute complexity as $1 -$ accuracy, where accuracy is averaged over all evaluated models. Higher values correspond to more challenging questions.

F Case Study

In this section, we provided qualitative examples to illustrate the reasoning trajectory generated by each method.

F.1 Failure Case 1

Table12 demonstrates the failure to distinguish between a necessary background condition and a causal difference-maker. While the model correctly identifies the black wire as a necessary condition (Step 4), it fails to apply the requisite world knowledge regarding causal selection. Specifically, the distinction between a standing "condition" and a precipitating "cause". In Step 5, the system explicitly articulates a flawed logical prior, asserting that any necessary but insufficient factor qualifies as a cause. Consequently, the model ignores the principle of normality, attributing causality to the component functioning within its design parameters (the black wire) rather than the deviating factor (the red wire), leading to an invalid conclusion.

Mathematical Reasoning Tasks
`Let's first understand the problem, extract relevant variables and their corresponding numerals, and make and devise a complete plan. Then, let's carry out the plan, calculate intermediate variables (pay attention to correct numerical calculation and commonsense), solve the problem step by step, and show the answer.`
COMMONSENSE TASKS
`Let's first prepare relevant information and make a plan. Then, let's answer the question step by step (pay attention to commonsense and logical coherence).`

Table 11: Prompts used for the Plan-and-Solve+ baseline under different task settings.

Table 12: An example of BBH causal reasoning. Incorrect causal inference is highlighted in red.

Prompt	***Question:*** A machine is set up in such a way that it will short circuit if both the black wire and the red wire touch the battery at the same time. The machine will not short-circuit if just one of these wires touches the battery. The black wire is designated as the one that is supposed to touch the battery, while the red wire is supposed to remain in some other part of the machine. One day, the black wire and the red wire both end up touching the battery at the same time. There is a short circuit. Did the black wire cause the short circuit? ***Options: (A) Yes (B) No***
ATAR	{ "Step 1: Understand the scenario": "The machine is designed to short circuit only when both the black wire and the red wire touch the battery at the same time.", "Step 2: Identify the necessary condition for the short circuit": "The necessary condition for the short circuit is that both the black wire and the red wire touch the battery at the same time.", "Step 3: Determine if the black wire's action was sufficient to cause the short circuit": "The black wire's action alone is not sufficient to cause the short circuit, as the machine will not short circuit if just one of these wires touches the battery.", "Step 4: Determine if the black wire's action was necessary to cause the short circuit": "The black wire's action was necessary to cause the short circuit, as the machine will not short circuit if just the red wire touches the battery.", "Step 5: Apply the concept of causation": "In causation, if the action of one thing is necessary for the occurrence of an event, but not sufficient on its own, then that thing is considered to have caused the event.", "Step 6: Determine if the black wire caused the short circuit": "Based on the necessary condition for the short circuit and the concept of causation, the black wire did cause the short circuit.", "Final answer": "(A) Yes" }

F.2 Failure Case 2

Table13 illustrates a representative instance of entity-attribute binding failure. The prompt specifies a relative increase in purchasing power ("5 more articles"). However, the model interpreted "5 more" as a fixed quantity of "6" (likely assuming she bought 1 before). This failure to distinguish between a marginal change and a total count results in the formulation of a fallacious linear equation, bypassing the necessary comparative logic required to solve for the original price.

F.3 Comparison Example 1

Table 14 illustrates a representative example where attention steering prevents reasoning errors. In Steps 2 and 4, the ATAR without attention steering approach incorrectly identifies "her" as referring to "the secretary"; this may be because models over-focus on sentence subjects rather than maintaining focus on the syntactic relationships that determine pronoun reference. In contrast, ATAR correctly identifies that "her" refers to "the analyst" by maintaining attention on both the original question context and the current reasoning step.

F.4 Comparison Example 2

In this example, both CoT (Table 15) and RE2 (Table 17) know English is the official language, but they drift into lengthy cultural descriptions and ultimately provide the wrong polarity. PS+ (Table 16) follows a reasonable reasoning path, but the final answer again diverges from the original question. ATAR (Table 18) maintains focus on the question and produces the correct answer.

G Numerical Experiment Result on Big-Bench-Hard

Table 13: An example of mathematical reasoning in Aqua. Incorrect numerical reasoning is highlighted in red.

Prompt	***Question:*** Anna is able to buy 5 more articles for $300 after the price of each article decreased by 15%. What is the new selling price of each article? ***Options: (A) $8 (B) $10 (C) $13.6 (D) $22.9 (E) $40***
ATAR	{ "Step 1: Establish variables": "Let the original price of each article be x.", "Step 2: Apply price decrease": "After a 15% decrease, the new price becomes $0.85x$.", "Step 3: Set up the cost equation": "Anna buys 5 more articles for $300, so the total cost is represented as $6 \times 0.85x = 300$.", "Step 4: Solve for the original price": "Solving the equation gives $x = \frac{300}{6 \times 0.85} = 58.82$.", "Step 5: Compute the new selling price": "The new price is $0.85 \times 58.82 = 50$.", "Step 6: Match with answer choices": "The value $50.00 does not appear in the options, leading to an incorrect selection based on proximity.", "Final answer": "(D) $22.9" }

Table 14: An example of BBH. Incorrect inferences are highlighted in red. Correct ones are highlighted in green.

Prompt	***Sentence: The secretary came to the analyst's office and helped her to book a flight.*** ***Options: (A) Helped the secretary book a flight (B) Helped the analyst book a flight (C) Ambiguous***
ATAR (w/o attention steering)	{ "Step 1: Identify the pronoun": "The pronoun is 'her'.", "Step 2: Determine the antecedent of the pronoun": "The antecedent of 'her' is the subject of the sentence, which is 'The secretary'", "Step 3: Analyze the sentence structure": "The subject of the sentence is 'The secretary', and the verb is 'came'. The object of the verb 'came' is 'to the analyst's office'. The verb 'helped' has the pronoun 'her' as its object.", "Step 4: Determine the antecedent of the pronoun based on the sentence structure": "Based on the sentence structure, the antecedent of the pronoun 'her' is the subject of the sentence, which is 'The secretary'.", "Step 5: Choose the correct option": "The correct option is (A) Helped the secretary book a flight."}
ATAR	{ "Step 1: Identify the pronoun": "The pronoun in the sentence is 'her'.", "Step 2: Determine the antecedent of the pronoun": "To find the antecedent, we need to look for the noun that 'her' refers to. In this case, 'her' is a possessive pronoun that refers to the analyst.", "Step 3: Analyze the sentence structure": "The subject of the sentence is 'The secretary', and the object of the verb 'helped' is 'her'", "Step 4: Identify the antecedent based on the sentence structure": "'her' refers to the analyst, not the secretary.", "Step 5: Choose the correct option": "Since 'her' refers to the analyst, the correct option is (B) Helped the analyst book a flight." }

Table 15: An example illustrating Chain-of-Thought's reasoning process.

Question:	Is the language used in Saint Vincent and the Grenadines rooted in English?
Chain-of-Thought (CoT)	No, the official language of Saint Vincent and the Grenadines is English, but the local Creole language, Saint Vincentian Creole, is also widely spoken. Saint Vincentian Creole is a English-based Creole language that incorporates elements of French, and African languages. It is the mother tongue of the majority of the population and is used in everyday conversation, music, and literature. While English is the official language and is used in government, education, and formal settings, Saint Vincentian Creole is the language of the people and is an important part of the country's cultural identity. So, while English is the official language, the local Creole language is also deeply rooted in the culture and daily life of Saint Vincent and the Grenadines. Final answer: No

Table 16: An example illustrating Plan-and-Solve's reasoning process.

Question:	Is the language used in Saint Vincent and the Grenadines rooted in English?
Plan-and-Solve (PS+)	Step 1: Understand the question The question asks about the language used in Saint Vincent and the Grenadines being rooted in English. This means we need to determine the primary language spoken in the country and its linguistic origins. Step 2: Research To answer this question, we need to research the languages spoken in Saint Vincent and the Grenadines. Saint Vincent and the Grenadines is an island nation in the southern part of the Windward Islands in the Caribbean Sea. Step 3: Analyze the findings After researching, we find that the official language of Saint Vincent and the Grenadines is English. However, the country also has a Creole language known as Vincentian Creole, which is a mix of English, French, and African languages. Step 4: Formulate the answer Given that the primary language used in Saint Vincent and the Grenadines is English, but the Creole language is also widely spoken and has roots in English, French, and African languages, the language used in Saint Vincent and the Grenadines is rooted in English to a significant extent. Step 5: Provide the final answer The answer is English, with Vincentian Creole, a mix of English, French, and African languages, also being widely spoken. Final answer: The final answer is English, with Vincentian Creole also being a significant part of the language landscape in Saint Vincent and the Grenadines.

Table 17: An example illustrating Re-Reading's reasoning process.

Question:	Is the language used in Saint Vincent and the Grenadines rooted in English?
Re-Reading (RE2)	No, the language used in Saint Vincent and the Grenadines is primarily English, but it also includes elements of French, African, and Indian languages due to its colonial history and diverse population. Final answer: No

Table 18: An example illustrating ATAR's reasoning process.

Question:	Is the language used in Saint Vincent and the Grenadines rooted in English?
ATAR	{"Understand the question.": " The question is asking whether the language spoken in Saint Vincent and the Grenadines is based on English.", "Research the official language of Saint Vincent and the Grenadines.": "Saint Vincent and the Grenadines is an island nation in the Caribbean, and its official language is English.", "Confirm the linguistic roots of the official language.": "English is a Germanic language, which means it has roots in Old English, a language spoken in parts of what is now England."} Final answer: Yes

Table 19: BBH task categories and the number of correct cases in each category.

Model	**Method**	**BBH**					
		disambiguation_qa	causal_judgement	date_understanding	logical_deduction	salient_translation_error_detection	snarks
Llama3.2-3B	CoT	88	71	102	34	95	89
	PS+	104	63	97	86	98	105
	re-read	74	84	127	40	89	105
	ToT	81	90	120	73	93	107
	ATAR	131	101	131	101	96	129
Phi4-mini-4B	CoT	162	113	154	114	149	134
	PS+	160	126	138	130	136	124
	re-read	162	115	165	113	135	148
	ToT	151	102	129	114	96	114
	ATAR	152	116	174	128	140	142
Qwen3-4B	CoT	183	121	174	208	167	148
	PS+	179	108	165	207	160	150
	re-read	179	121	203	230	147	154
	ToT	172	114	176	219	122	150
	ATAR	185	115	193	224	161	150
Llama3.1-8B	CoT	100	91	182	67	110	125
	PS+	138	108	136	78	124	122
	re-read	124	98	190	85	121	127
	ToT	114	104	153	87	120	111
	ATAR	152	100	160	128	132	127
Phi-4-15B	CoT	180	120	203	169	152	160
	PS+	180	117	140	186	155	156
	re-read	179	120	221	191	158	154
	ToT	145	90	198	147	145	150
	ATAR	176	122	218	203	148	161
Qwen3-30B	CoT	175	125	182	185	172	155
	PS+	184	122	172	181	165	156
	re-read	183	125	185	210	174	156
	ToT	180	132	195	186	158	153
	ATAR	180	123	196	200	160	157