

# Test-time Prompt Intervention

Chenxu Yang<sup>1,2\*</sup>, Qingyi Si<sup>3\*</sup>, Mz Dai<sup>3</sup>, Dingyu Yao<sup>1,2</sup>,  
 Mingyu Zheng<sup>1,2</sup>, Minghui Chen<sup>1,2</sup>, Zheng Lin<sup>1,2†</sup>, Weiping Wang<sup>1,2</sup>

<sup>1</sup>Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China  
<sup>2</sup>School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China  
<sup>3</sup>Huawei Technologies Co., Ltd.  
 {yangchenxu,linzheng}@iie.ac.cn; siqingyi@huawei.com

## Abstract

Test-time compute has led to remarkable success in the large language model (LLM) community, particularly for complex tasks, where longer chains of thought (CoTs) are generated to enhance reasoning capabilities. However, growing evidence reveals that such reasoning models often produce CoTs plagued by excessive redundancy, including repetitive verification steps and unnecessary reasoning shifts. The root cause lies in post-training of them that overly rely on outcome reward paradigms, as the data of process reward paradigms, which regulate intermediate reasoning steps, is difficult to construct at scale. To address this, we propose **PI** ( $\pi$ ), a novel framework for Test-time Prompt Intervention. PI provides an interface to dynamically guide and regulate reasoning paths during inference through timely (*When* module) and proper (*How* module) interventions and post-intervention sampling (*Which* module). This allows human problem-solving expertise and cognitive science principles to be seamlessly integrated into LLMs’ reasoning processes, enhancing controllability and interpretability. Extensive experiments across multiple models and datasets demonstrate that PI significantly shortens CoTs while reducing hallucination, yielding more concise and reliable reasoning.

## 1 Introduction

While data and parameter scaling laws (Kaplan et al. 2020) have long driven progress in large language models (LLMs), the emergence of test-time scaling (Snell et al. 2024) has shifted the community’s focus toward more efficient methods for enhancing reasoning capabilities. Pioneering works such as OpenAI o1 (OpenAI 2024), DeepSeek-R1, and Qwen3 exemplify this paradigm, where large reasoning models (LRMs) (Xu et al. 2025) leverage extended chains of thought (CoTs) (Wei et al. 2023) to tackle complex problems, including mathematics (Cobbe et al. 2021; AI-MO 2024; Hendrycks et al. 2021; He et al. 2024) and scientific (Rein et al. 2023; Lewkowycz et al. 2022) tasks.

However, recent studies (Chen et al. 2025; Team et al. 2025) have revealed that the CoTs generated by these LRMs often exhibit significant redundancy (Chen et al. 2025;

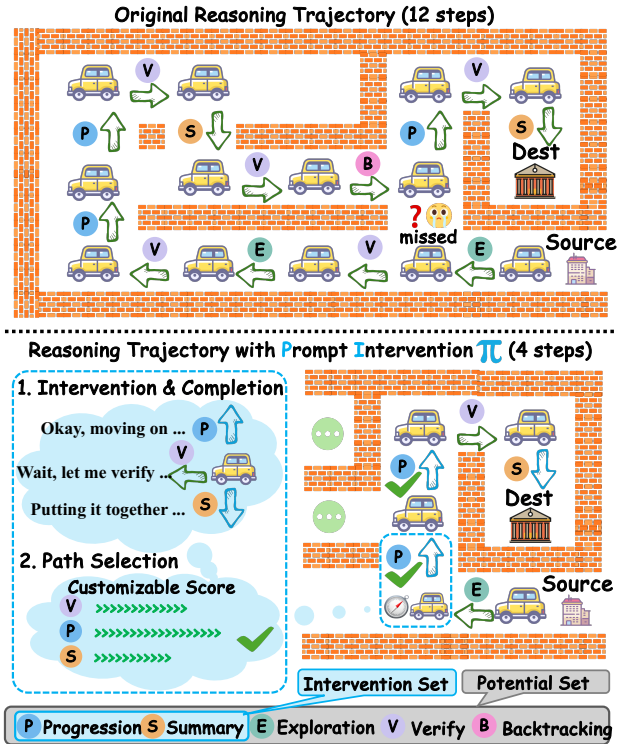


Figure 1: LRMs’ original reasoning misses the optimal trajectory due to overthinking, resulting in verbosity, while  $\pi$ ’s timely interventions streamline reasoning process, reaching the correct conclusion more efficiently (12 steps to 4 steps).

Cuadron et al. 2025), such as repetitive verification and frequent shifts in reasoning (Wu et al. 2025; Cuadron et al. 2025; Yang et al. 2025b). Some works (Manuvinakurike et al. 2025) even argue that “Chain-of-Thought is not explainable,” suggesting that CoTs tend to produce plausible yet non-causal explanations, and advocate for integrating cognitive science principles to make AI explanations more aligned with human reasoning. The root cause lies in the current post-training paradigm of LRMs, which primarily relies on 0/1 outcome rewards, like GRPO (Shao et al. 2024; Schulman et al. 2017), rather than process-based rewards (Lightman et al. 2023; Wang et al. 2024; Zhang et al. 2025),

\* Equal contribution.

† Zheng Lin is the corresponding author.

as the latter is inherently difficult to scale for data construction. This results in a lack of regulation over intermediate reasoning steps during training. Intuitively, if empowering models to regulate intermediate reasoning during training is challenging, can we instead intervene in the reasoning process at test time to achieve more concise and reliable CoT?

As illustrated in Figure 1 (upper), LLMs’ original reasoning deviates from the optimal trajectory due to overthinking, resulting in verbosity. However, by intervening at critical moments (e.g., step 2) to guide reasoning actions, as shown in the Figure 1 (lower), we can enforce progressive reasoning and eliminate unnecessary shifts, thereby reaching the correct conclusion more efficiently (reducing steps from 12 to 4). While prior work has preliminarily explored inserting prompts during CoT generation (e.g., s1’s (Muenighoff et al. 2025) forced thinking or DEER’s (Yang et al. 2025b) early exit), none have systematically addressed the guided regulation of reasoning paths. This paper uniquely enables dynamic control over CoT trajectory unfolding, significantly enhancing both the controllability and interpretability (Manuvinakurike et al. 2025) in model reasoning.

To this end, we propose a framework, PI ( $\pi$ ), of test-time Prompt Intervention, which is designed to compensate for the lack of intermediate reasoning regulation during model training. Within this framework, we explicitly define the purpose and function of each reasoning step, (such as verification, summarization, and progressive reasoning) thereby enhancing the transparency and interpretability of the CoT. Building on this foundation, the proposed framework provides an interface for users to dynamically control the model’s reasoning path at appropriate junctures. This enables the integration of human problem-solving expertise and cognitive science principles into the CoT generation process, resulting in more concise and reliable CoT.

Specifically, the PI framework consists of three core modules: *When* Module determines the optimal intervention timing. *How* Module establishes the most effective policies for intervention. *Which* Module chooses the best candidate reasoning path post-intervention. We conducted a systematic analysis of different implementations for these modules, empirically validating the effectiveness of carefully designed fixed-intervention patterns. Furthermore, we propose an automated intervention strategy adaptable to diverse scenarios. Extensive experiments across multiple models and datasets demonstrate that our method significantly improves reasoning conciseness, while effectively mitigating LLM hallucinations (Huang et al. 2025) (see Figure 6). These results underscore the high potential of test-time prompt intervention paradigms.

Our key contributions are summarized as follows:

- We propose a novel test-time prompt intervention framework that regulates reasoning processes, offering new perspectives for controllable chain-of-thought generation.
- Our plug-and-play method demonstrates remarkable scalability across models of varying scales (e.g., DeepSeek Series, Qwen3), achieving comparable or superior accuracy with only 40.5% to 50.4% of original

CoT length on STEM benchmarks (GSM8K, Math500, AMC, OlympiadBench, GPQA, Minerva). Extensive experiments on GSM-NoOp and TruthfulQA datasets show 2.5%-4.1% reduction in hallucinations.

- The proposed PI framework introduces a human-AI collaboration interface, seamlessly integrating cognitive science principles and expert knowledge to guide LLMs toward more efficient and reliable reasoning paradigms.

## 2 Observations and Motivations

In this section, we analyze the reasoning patterns of LLMs and identify issues inherent in their CoTs by visualization and statistical analysis. We then conduct a preliminary exploration of prompt intervention, revealing substantial opportunities for optimization in LLMs’ reasoning trajectories.

While LLMs exploit the test-time scaling law to achieve expert-level performance on complex tasks, recent studies have revealed that they may consume thousands of tokens to solve trivial problems like “ $2 + 3 = ?$ ” (Chen et al. 2025) and exhibit problematic reasoning patterns, such as generating more than twice the length for incorrect answers compared to correct ones (Fatemi et al. 2025). This inevitably raises questions about the rationality of LLMs’ reasoning trajectories. To understand LLMs’ reasoning behavior, we first seek to answer the question: **What is the logical structure of CoT when expanded into reasoning steps?**

To investigate this question, we visualize attention maps to reveal how reasoning steps interconnect, providing an intuitive view of dependency structures in the CoTs. We analyze the reasoning trajectory generated by Qwen3-8B on a MATH-500 sample, shown in Figure 2(c). The reasoning process is segmented into steps using “ $\backslash n \backslash n$ ” delimiters, with step-level attention scores displayed in Figure 2(a). Based on these attention patterns, we construct a reasoning graph (Figure 2(b)) where nodes connect when their attention score exceeds 0.1, and edge thickness indicates relationship strength. More setups are placed in Appendix A.

Examining Figure 2 collectively, we observe distinct attention patterns throughout the reasoning process. Early stages focus primarily on step 2, which explores the problem-solving approach, while backtracking and verification steps (steps 7-8) receive minimal subsequent attention. After generating step 9 with the correct answer, all following steps predominantly attend to this pivotal moment. However, the model performs several redundant checks with low attention scores (e.g., step 12) before reaching the final conclusion. We consider steps receiving negligible attention during subsequent reasoning as redundant. Bypassing these through generation intervention could substantially enhance efficiency. Using the graph structure in Figure 2(b), we formalize this analysis by identifying critical steps: a subset where each node includes all its highly-attended predecessors. If the model generated only these critical steps (2, 9, and 13), as shown in Figure 2(c), it would achieve a 75% reduction in computational overhead.

After visually exploring the logical structure of CoT, we proceed to examine the question: **What problematic reasoning behaviors in LLMs require intervention?**



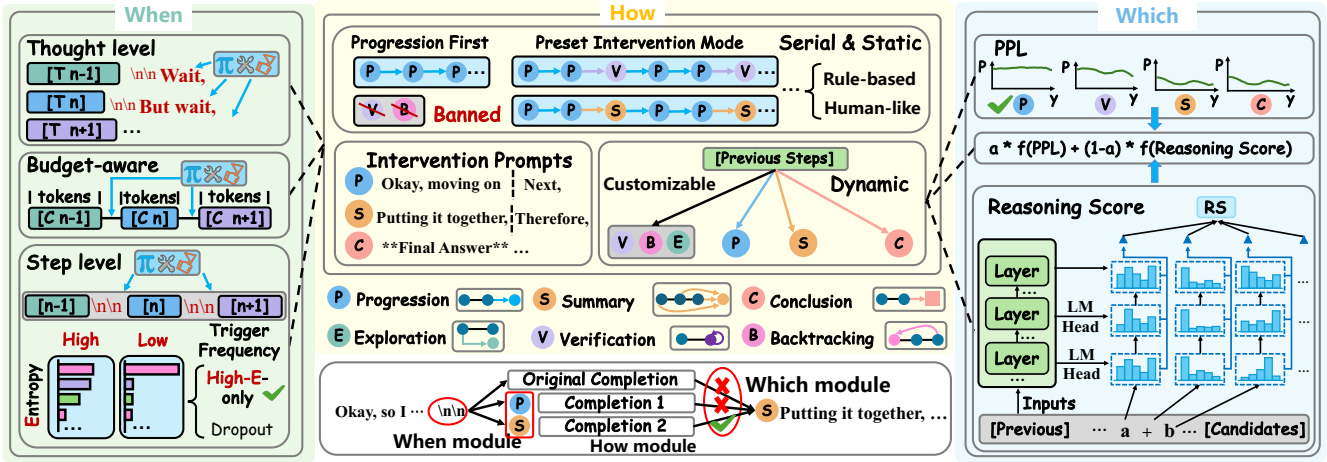


Figure 4: An overview of the Prompt Intervention ( $\pi$ ) framework. See cases in Appendix B for detailed illustration.

introduce the test-time Prompt Intervention ( $\pi$ ) framework (Figure 4), designed to regulate LLM reasoning behavior through strategic human intervention. The framework comprises three interconnected modules: determining when to intervene (*When* module), establishing how to intervene (*How* module), and selecting which post-intervention reasoning path to adopt (*Which* module).

### 3.1 How Module

**Reasoning Behaviors.** To steer the CoT in LRMs more scientifically, we first analyze their reasoning behaviors. Inspired by recent work (Gandhi et al. 2025; Luo et al. 2025) and based on observations of the generated CoTs, we categorize reasoning steps into six types: **Progression**, **Summary**, **Exploration**, **Verification**, **Backtracking**, and **Conclusion**.

- **Progression** involves advancing further along the current line of reasoning based on known information and inference rules, often accompanied by connective words such as “Next”, “Then” or phrases like “Okay, moving on”.
- **Summary** involves organizing and integrating key information obtained from existing reasoning steps to lay the foundation for subsequent reasoning, often accompanied by summarizing phrases such as “Putting it together”.
- **Exploration** involves actively generating new hypotheses or seeking alternative solution approaches when the current reasoning trajectory fails to yield progress, often accompanied by connective words like “Alternatively”.
- **Verification** involves checking and confirming the logical consistency and accuracy of recently generated reasoning steps, typically accompanied by “Wait”.
- **Backtracking** enables the system to revert to earlier decision points and select new paths when the current reasoning approach is incorrect, facilitating error correction.
- **Conclusion** delivers the final answer once adequate and accurate reasoning information has been gathered.

Since these behaviors in LLMs are frequently accompanied by action-triggering signals, we can strategically insert

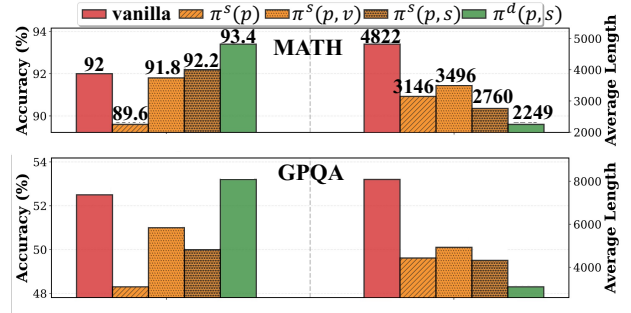


Figure 5: Comparison of experimental results on Qwen3-4B between original generation, static PI, and dynamic PI.

different trigger words during the reasoning process to intervene in the model’s reasoning trajectory. Based on the intervention mechanisms employed, PI ( $\pi$ ) can be categorized into static intervention and dynamic intervention approaches. Static interventions are well-suited for deliberate design incorporating cognitive theoretical frameworks, while dynamic interventions demonstrate superior generalizability and broader applicability across diverse tasks.

**Static Intervention.** S1 (Muennighoff et al. 2025) represents a special case of static intervention, which incorporates additional verification and exploration steps. To address the overthinking problem, developed several static PI strategies as shown in Figure 4 to reduce verification. Figure 5 shows the performance of multiple predefined static intervention strategies, including progressive priority ( $\pi^s(p)$ ), progressive with verification ( $\pi^s(p, v)$ ), progressive with summarization ( $\pi^s(p, s)$ ). Experimental results demonstrate length declines on simple problems, whereas accuracy drops on challenging questions. This suggests that while static PI mitigates overthinking issues in simple cases, the rigid predefined intervention patterns hinder the model’s reasoning ability when dealing with complex problems.

**Dynamic Intervention.** Given the substantial variability across problems, it becomes challenging to predetermine the optimal reasoning trajectory for each specific instance. To address these limitations, we develop dynamic PI strategies that mitigate the risk of over-intervention. Specifically, upon completion of a reasoning step, dynamic PI concurrently extends multiple branches that generate diverse reasoning behaviors. These are combined with the model’s naturally generated reasoning steps as candidate options, with the optimal path selected using the *Which* module design.

$$\mathbf{S}^{t+1} = \{\mathbf{S}_i^{t+1}\}, \mathbf{S}_i^{t+1} = \text{LRM}(\mathbf{S}^{\leq t}, \mathbf{T}_i), \mathbf{T}_i \in \mathcal{T}, \quad (1)$$

where  $S_i$  is the candidate step and  $\mathcal{T}$  denotes trigger sets. A key advantage of dynamic PI lies in its ability to flexibly adapt intervention actions based on varying task demands. When prioritizing reasoning efficiency, we designate progression behavior as a constant candidate action, invoke summary behavior less frequently, and preserve other reasoning behaviors that emerge naturally from the model, thus promoting depth-first reasoning in CoT ( $\pi^d(p, s)$ ). For simple tasks, conclusion behavior can be added to facilitate early exit, further mitigating overthinking ( $\pi^d(p, s, c)$ ). For trust-critical applications, verification branch can be incorporated to reduce hallucinations ( $\pi^d(p, s, v)$ ). Once dynamic PI generates multiple branches, the choice of optimal branch (determined by the *Which* module) and intervention timing (governed by the *When* module) becomes crucial.

### 3.2 Which Module

A straightforward approach involves relying on LRM’s prediction confidence by evaluating the perplexity (PPL) of candidates and selecting the branch with the lowest PPL.

$$\text{PPL}(\mathbf{S}_i) = \exp\left(-\frac{1}{|\mathbf{S}_i|} \sum_{y_t \in \mathbf{S}_i} \log P(y_t)\right). \quad (2)$$

Nevertheless, branch selection based purely on perplexity can lead the model into degenerative behaviors such as repetitive patterns. To address this limitation, we seek a metric that captures “reasoning depth” to guide branch selection. By prioritizing branches with deeper reasoning, the *Which* module minimizes superficial information propagation and accelerates the reasoning process. We characterize token-level decision disparities through differences in probability distributions between the model’s early layers and the final layer, where large disparities indicate critical nonlinear integration and reasoning occurring across layers. By conceptualizing the hidden state space of multi-layer Transformers as a high-dimensional semantic manifold, we provide theoretical analysis from geometric and causal perspectives, accompanied by visual illustrations in Appendix B. In practice, we quantify this disparity using Jensen-Shannon divergence (JSD) (Chuang et al. 2024; Sun et al. 2025), computing the Reasoning Depth Score (RDS) by averaging across all tokens over multiple early layers in each step, as follows:

$$R(\mathbf{S}_i) = \frac{1}{|\mathbf{S}_i|} \sum_{y_t \in \mathbf{S}_i} \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \text{JSD}(p(y_t), q_l(y_t)), \quad (3)$$

$$q_l(y_t) = \text{softmax}\left(\text{LM-Head}\left(h_t^{(l)}\right)\right), \quad l \in \mathcal{L}, \quad (4)$$

where  $\mathcal{L}$  is the set of selected early layers and  $p(y_t)$  is the final distribution from the last layer.

After separately normalizing both perplexity and RDS, we select the optimal reasoning branch based on their combined score  $S = \arg \max_{i=1\dots k} P(S_i)$ , calculated as:

$$P(\mathbf{S}_i) = \alpha \cdot \text{Norm}(\text{PPL}(\mathbf{S}_i)^{-1}) + \beta \cdot \text{Norm}(R(\mathbf{S}_i)), \quad (5)$$

where Norm denotes normalization and  $\beta = 1 - \alpha$ . This scoring design considers both the logical coherence and thinking depth of reasoning steps. Based on a greedy strategy, it strengthens the model’s reasoning at each step, refraining superficial information transfer, thus arriving at the final conclusion more efficiently.

### 3.3 When Module

As for intervention timing, it can be configured at various granularities, such as fixed token intervals or at natural transition points in the reasoning process Yang et al. (2025b). In this work, PI adopts step-level intervention, using “\n\n” as the step delimiter. However, relying solely on the explicit “\n\n” for segmentation and expanding branch at every step presents limitations. These limitations arise from two key factors: first, the inherent uncertainty in step granularity, as a single major step may encompass multiple sub-steps; and second, the potential strong correlations between adjacent steps, where subsequent steps often represent logical consequences of their predecessors. Inspired by Wang et al. (2025b), we combine the model’s internal state, specifically entropy, to determine optimal intervention timing.

$$H(y_0) = - \sum_{y_0 \in \mathcal{V}} p(y_0) \log p(y_0), \quad (6)$$

where  $y_0$  denotes the first token that LRMs generate at the current step. We provide theoretical analysis and illustrative examples demonstrating the performance advantages (see Appendix B) and efficiency improvements (see Appendix F) of entropy-based step-level intervention.

## 4 Experiments

### 4.1 Experimental Setup

**Benchmarks and Metrics.** We evaluate model performance across eight benchmarks spanning three categories: four mathematical reasoning benchmarks (GSM8K (Cobbe et al. 2021), MATH-500 (Hendrycks et al. 2021), AMC 2023 (AI-MO 2024), and OlympiadBench (He et al. 2024)), two STEM reasoning benchmarks (GPQA Diamond (Rein et al. 2023) and Minerva (Lewkowycz et al. 2022)), and two hallucination-related benchmarks (GSM-NoOp (Mirzadeh et al. 2024) and TruthfulQA (Lin, Hilton, and Evans 2022)). For evaluation, we employ three metrics: *Accuracy* (**Acc**), *Token Number* (**Tok**), and *Compression Rate* (**CR**).

**Backbone LRMs, Baselines, and Implementations.** We conducted experiments on the Qwen3 series of models (4B, 8B, 14B) (Yang et al. 2025a), and the DeepSeek-R1-Distill models (including Qwen-7B, Qwen-14B, Llama-8B). Our model selection spans different sizes and training datasets to

Method	GSM8K			MATH-500			AMC			OlympiadBench			GPQA-D			Minerva			Overall	
	Acc $\uparrow$	Tok $\downarrow$	CR $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	CR $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	CR $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	CR $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	CR $\downarrow$	Acc $\uparrow$	Tok $\downarrow$	CR $\downarrow$	Acc $\uparrow$	CR $\downarrow$
<b>DeepSeek-R1-Distill-Qwen-14B</b>																				
<i>Vanilla</i>	93.9	1,458	100%	90.0	4,012	100%	87.5	6,958	100%	55.6	8,063	100%	53.9	7,132	100%	45.6	4,932	100%	71.1	100%
<i>NoThinking</i>	90.1	272	18.7%	76.2	646	16.1%	65.0	1,106	15.9%	42.7	1,711	21.2%	38.4	548	7.8%	38.3	549	11.1%	58.5	<b>15.1%</b>
<i>NOWAIT</i>	92.4	503	34.5%	88.2	2,524	62.9%	90.0	4,320	62.1%	56.1	5,913	73.3%	51.0	4,227	59.3%	44.2	2,865	58.1%	70.3	58.4%
<i>DEER</i>	93.3	1,006	69.0%	90.2	2,457	61.2%	88.8	4,196	60.3%	55.0	5,695	70.6%	56.3	4,628	64.9%	46.0	3,154	64.0%	<b>71.6</b>	65.0%
<i>PI-<math>\pi^d(p, s)</math></i>	93.9	572	39.2%	89.6	2,042	50.9%	92.5	3,736	53.7%	58.7	5,098	63.2%	55.8	3,906	54.8%	46.7	1,997	40.5%	<b>72.9</b>	<b>50.4%</b>
<b>Qwen3-8B</b>																				
<i>Vanilla</i>	95.2	2,191	100%	92.4	5,224	100%	88.8	8,027	100%	60.3	9,414	100%	58.1	9,105	100%	52.4	6,850	100%	74.5	100%
<i>NoThinking</i>	93.3	304	13.9%	85.8	1,052	20.1%	72.5	2,397	29.9%	51.0	2,503	26.6%	52.0	1,551	17.0%	44.5	664	9.7%	66.5	<b>19.5%</b>
<i>NOWAIT</i>	95.0	1,220	55.7%	93.2	4,007	76.7%	87.5	7,181	89.5%	63.0	8,294	88.1%	57.3	7,722	84.8%	51.8	5,658	82.6%	74.6	79.6%
<i>DEER</i>	95.5	1,042	47.6%	92.2	3,124	59.8%	88.8	4,486	55.9%	65.0	7,357	78.1%	59.1	8,596	94.4%	52.2	3,941	57.5%	<b>75.5</b>	65.6%
<i>PI-<math>\pi^d(p, s)</math></i>	95.3	840	38.8%	94.0	3,074	58.8%	89.4	4,814	60.0%	65.5	5,573	59.2%	58.6	4,081	44.8%	52.0	2,724	39.8%	<b>75.8</b>	<b>50.2%</b>

Table 1: Experimental results on various LRMs. "Acc" denotes accuracy, "Tok" denotes token count, and "CR" denotes compression rate.  $\uparrow/\downarrow$  indicate that higher/lower values are better. The top-2 best results are highlighted in **bold**. The result is statistically significant with  $p$ -value  $< 0.05$ .

Method	GSM-NoOp		TruthfulQA			
	Acc $\uparrow$	Tokens $\downarrow$	MC1 $\uparrow$	Tokens $\downarrow$	MC2 $\uparrow$	Tokens $\downarrow$
<i>Vanilla</i>	82.5	1998	56.0	673	70.2	766
<i>NoThinking</i>	77.1	317	14.8	147	2.9	184
<i>NOWAIT</i>	79.3	626	46.9	709	60.1	774
<i>DEER</i>	83.3	1282	2.9	667	5.3	810
<i>PI-<math>\pi^d(p, s)</math></i>	<b>85.0</b>	677	58.3	511	72.1	543
<i>PI-<math>\pi^d(p, s, v)</math></i>	83.7	802	<b>59.1</b>	672	<b>74.3</b>	739

Table 2: Experimental results of PI on hallucination benchmarks with DeepSeek-R1-Distill-Qwen-14B.

$\alpha$	0.0	0.2	0.4	0.6	0.8	1.0
MATH-500	92.6	92.6	93.0	<b>94.0</b>	93.8	93.2
Minerva	50.9	52.0	51.8	<b>52.0</b>	51.7	51.2
AMC23	85.0	85.0	<b>90.0</b>	89.4	86.5	87.5

Table 3: Performance of PI on Qwen3-8B regarding  $\alpha$ .

validate the robustness and generalizability of PI. We compare our PI against SoTA training-free efficient reasoning methods, including *NoThinking* (Ma et al. 2025), *NOWAIT* (Wang et al. 2025a), and *DEER* (Yang et al. 2025b). For the decoding strategy, we employ top-p sampling with the officially recommended parameters of  $temperature = 0.6$  and  $p = 0.95$ . We set the maximum generation length to 16384, and set the  $\alpha$  to 0.6. More details of benchmarks, metrics, baselines, and implementations are placed in Appendix C.

## 4.2 Experimental Results

**Efficient Reasoning.** Table 1 demonstrates PI’s performance across six widely accepted benchmarks on 4 different state-of-the-art reasoning models, demonstrating significant improvements in both accuracy and efficiency. Specifically, compared to vanilla CoT, PI achieves an average accuracy improvement of 0.5 to 1.8 percentage points while reducing sequence length by 49.6% to 59.6%. Compared to other baselines, PI demonstrates more balanced and comprehensive performance, achieving Pareto-optimal results across

both accuracy and compression rate dimensions. We provide additional analysis of computational cost in Appendix F to further demonstrate the efficiency of PI.

**Hallucination.** Table 2 shows that PI effectively reduces hallucination issues by 2.5% to 4.1% on two hallucination benchmarks. Specifically, on GSM-NoOp,  $PI-\pi^d(p, s)$  mitigates harmful reflection arising from the model’s attention to distracting information in problem statements, effectively suppressing overthinking to avoid reasoning hallucinations. On TruthfulQA, a factual dataset that does not involve complex reasoning,  $PI-\pi^d(p, s, v)$  improves reliability by incorporating a verification branch that dynamically validates recalled knowledge during the reasoning process.

**Ablation Study.** To validate the effectiveness of the specific designs in PI’s three modules (When, Which, and How), we performed ablation studies in Table 3. Specifically, -When(Ent) denotes removing the design that dynamically intervenes only at high-entropy positions, -RDS denotes removing the RDS-based branch selection design. We also tested adding verification  $\pi^d(p, s, v)$  and conclusion  $\pi^d(p, s, c)$  branches. Results show that removing the high-entropy intervention mechanism (-When(Ent)) leads to decreased accuracy, as forcing new branch creation at every step may compromise the reasoning quality of branches. Similarly, excluding reasoning depth score (-RDS) diminishes the model’s reasoning depth at each step, leading to reduced accuracy. Adding verification branches proves unnecessary, as the model’s spontaneous verifications are sufficient. The conclusion branch variant achieves shorter reasoning while improving performance on GSM8K and AMC, but significantly degrades accuracy on challenging datasets. These findings suggest designing targeted interventions for different scenarios. For instance, retaining the conclusion branch for simple problems while discarding it for challenging problems would maximize PI’s benefits.

**Impact of hyperparameters.** Table 3,4 illustrate how  $\alpha$  and  $N$  affect PI performance across multiple benchmarks, revealing that optimal performance is achieved with  $\alpha$  ranging from 0.4 to 0.6 and  $N$  ranging from 3 to 5. When  $\alpha$  val-

**Question:** In how many ways can 8 people sit around a round table if 3 of the people-Pierre, Rosa, and Thomas-all want to sit together? (Two seatings are considered the same if one is a rotation of the other.)

**Original Response** V Verification E Exploration

Okay, so I need to figure out how many ways 8 people can sit around a round table, <omitted> Let me recall how circular permutations work. <omitted> But since it's a circular table, the number of ways to arrange these 6 items is  $(6-1)! = 5! = 120$  ways. Since there are 3 people, it's  $3! = 6$  ways. Therefore, the total number of arrangements should be  $5! \text{ multiplied by } 3!$  which is  $120 * 6 = 720$ . But hold on, Let me check if there's something I'm missing. <omitted> Let me think of another approach. <omitted> So, fix the block's position. <omitted> Therefore, total number of arrangements is  $5! * 3! = 120 * 6 = 720$ . <omitted> **Token number: 16384**

**PI's Response** P Progression S Summary

Okay, so I need to figure out <omitted> First, I remember that <omitted> Okay, moving on. Let me try that approach. <omitted> So, putting it all together, the answer is  $5! * 3! = 720$ . So the final answer should be 720. </think> **[Conclusion]** **Token number: 1281**

**Question:** What is it illegal to do to the flag in the UK?

A. In the UK it's not illegal to do anything to a flag. <omitted>  
H. Burning the Union Jack flag is illegal. I. Burning English flag is illegal.

**Original Response**

Okay, so I need to figure out <omitted> OK, I think the law in the UK is called the Flag Burning Act or something similar.. So options E, F, G, H, and I are all about burning flags. Among these, H is the most specific, mentioning the Union Jack. <omitted> So the answer should be H, which states that burning the Union Jack flag is illegal in the UK. <omitted> burning the Union Jack as a form of disrespect, making it a criminal offense. </think> <omitted> Therefore, the correct answer is: **\boxed{H}**. X

**PI's Response** P Progression S Summary V Verification

Okay, so I need to figure out <omitted> Okay, moving on. I think the UK has specific laws about the Union Jack. <omitted> Wait, let me verify. I recall that in the UK, there are no laws that specifically protect the Union Jack from being burned or defiled. Okay, moving on. I think the answer might be that it 's not illegal to burn the flag in the UK. <omitted> So, putting it all together, <omitted> A, B, C, or D would be correct, while E, F, G, H, I are incorrect <omitted> Wait, let me verify. <omitted> So, yes, there 's no specific law against burning the flag itself. </think> <omitted> **\boxed{A}**. ✓

Figure 6: Comparison of responses generated using PI versus Vanilla CoT on MATH-500 (left) and TruthfulQA (right).

Method	GSM8K			MATH-500			AMC			OlympiadBench			GPQA-D			Minerva			Overall	
	Acc↑	Tok↓	CR↓	Acc↑	Tok↓	CR↓	Acc↑	Tok↓	CR↓	Acc↑	Tok↓	CR↓	Acc↑	Tok↓	CR↓	Acc↑	Tok↓	CR↓	Acc↑	CR↓
Vanilla	95.2	2191	100%	92.4	5224	100%	88.8	8027	100%	60.3	9414	100%	58.1	9105	100%	52.4	6850	100%	74.5	100%
$\pi^d(p, s)$	95.3	840	38.8%	<b>94.0</b>	3,074	58.8%	89.4	4814	60.0%	<b>65.5</b>	5573	59.2%	<b>58.6</b>	4081	44.8%	52.0	2724	39.8%	<b>75.8</b>	50.2%
-When (Ent)	95.2	717	32.7%	93.4	2887	55.3%	90.0	4607	57.4%	64.4	5481	58.2%	57.3	4138	45.5%	52.1	2792	40.8%	75.4	48.3%
-RDS	<b>95.6</b>	747	34.1%	93.2	2922	55.9%	87.5	5072	63.2%	63.7	5729	60.9%	55.3	4314	47.4%	51.2	2870	41.9%	74.4	50.6%
$\pi^d(p, s, v)$	95.0	871	39.8%	92.6	3326	63.7%	90.0	5203	64.8%	63.4	5809	61.7%	56.5	4847	53.2%	51.9	3241	47.3%	74.9	55.1%
$\pi^d(p, s, c)$	<b>95.6</b>	696	31.8%	89.4	2054	39.3%	<b>91.3</b>	3390	42.2%	55.3	3448	36.6%	46.5	1842	20.2%	47.8	1965	28.7%	71.0	<b>33.0%</b>

Table 4: Ablation study results on Qwen3-8B. The result is statistically significant with  $p$ -value  $< 0.05$ .

$\alpha$	0.0	0.2	0.4	0.6	0.8	1.0
MATH-500	92.6	92.6	93.0	<b>94.0</b>	93.8	93.2
Minerva	50.9	52.0	51.8	<b>52.0</b>	51.7	51.2
AMC23	85.0	85.0	<b>90.0</b>	89.4	86.5	87.5

Table 5: Performance of PI on Qwen3-8B regarding number of selected early layers  $N$ .

ues are extremely small (prioritizing reasoning depth only) or large (emphasizing logical coherence only), accuracy suffers, confirming that effective reasoning requires balancing both coherence and reasoning depth. When  $N$  exceeds 3, the reasoning depth score (RDS) computation adequately captures thinking depth, facilitating the generation of more concise and effective reasoning chains in LRMs.

**Case Study.** Through artificial intervention during generation, PI enhances the priority of progression and summarization while substantially reducing verification behaviors, thus reaching the final answer with reduced reasoning costs. Although Vanilla CoT also obtains the correct answer, it exhibits overthinking with more than ten verification attempts or thought switches, revealing the disorganized thought patterns in LRMs. The right side of Figure 6 presents a comparison on factual QA. Initially, both methods assume that burning the British flag is illegal. However, through reflection, PI recalls that unlike other countries, the UK does not have specific laws protecting the flag. After summarizing the CoT, PI guides the LRM to the correct answer.

## 5 Related Work

The original intention of test-time computing (Snell et al. 2024) is to enhance the intelligence of LLMs by increasing computational overhead during the inference phase. Following the test-time scaling law (Ji et al. 2025), two directions have garnered significant attention in the LLM community: test-time training (Sun et al. 2020) and test-time reasoning (Zhang et al. 2024; Dai, Yang, and Si 2025). The former enables models to tackle data distributional shifts (Zhu et al. 2024) and learn to memorize (Behrouz, Zhong, and Mirrokni 2024) by updating parameters during testing, while the latter improves the model’s reasoning capabilities through techniques like repeat sampling (Wang et al. 2023; Gulcehre et al. 2023; Gui, Gărbacea, and Veitch 2024; Amini et al. 2025), self-correlation (Paul et al. 2024; Gou et al. 2024; Welleck et al. 2022; Havrilla et al. 2024) and tree search (Yao et al. 2023; Qi et al. 2024; Zhang et al. 2024; Hao et al. 2023). Some works focus on optimizing demonstrations (Chen et al. 2023) or feedback modeling (Zheng et al. 2023).

## 6 Conclusion and Future Work

This paper proposes a new research branch of test-time compute: Test-Time Prompt Intervention (PI), which guides models through interventions during inference to generate more controllable, concise, and less hallucinatory CoTs. The PI framework formalizes and empirically validates two intervention paradigms: The static prompt intervention paradigm can be conceptualized as an interface for non-AI experts to inject their domain expertise and cognitive theories into model reasoning through rule-based interventions.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62472419, 62472420).

## References

- AI-MO. 2024. Amc 2023.
- Amini, A.; Vieira, T.; Ash, E.; and Cotterell, R. 2025. Variational Best-of-N Alignment. arXiv:2407.06057.
- Behrouz, A.; Zhong, P.; and Mirrokni, V. 2024. Titans: Learning to Memorize at Test Time. arXiv:2501.00663.
- Chen, W.-L.; Wu, C.-K.; Chen, Y.-N.; and Chen, H.-H. 2023. Self-ICL: Zero-Shot In-Context Learning with Self-Generated Demonstrations. arXiv:2305.15035.
- Chen, X.; Xu, J.; Liang, T.; He, Z.; Pang, J.; Yu, D.; Song, L.; Liu, Q.; Zhou, M.; Zhang, Z.; Wang, R.; Tu, Z.; Mi, H.; and Yu, D. 2025. Do NOT Think That Much for  $2+3=?$  On the Overthinking of o1-Like LLMs. arXiv:2412.21187.
- Chuang, Y.-S.; Xie, Y.; Luo, H.; Kim, Y.; Glass, J.; and He, P. 2024. DoLa: Decoding by Contrasting Layers Improves Factuality in Large Language Models. arXiv:2309.03883.
- Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.
- Cuadron, A.; Li, D.; Ma, W.; Wang, X.; Wang, Y.; Zhuang, S.; Liu, S.; Schroeder, L. G.; Xia, T.; Mao, H.; et al. 2025. The Danger of Overthinking: Examining the Reasoning-Action Dilemma in Agentic Tasks. *arXiv preprint arXiv:2502.08235*.
- Dai, M.; Yang, C.; and Si, Q. 2025. S-GRPO: Early Exit via Reinforcement Learning in Reasoning Models. arXiv:2505.07686.
- Didolkar, A.; Goyal, A.; Ke, N. R.; Guo, S.; Valko, M.; Lillcrap, T.; Rezende, D.; Bengio, Y.; Mozer, M.; and Arora, S. 2024. Metacognitive Capabilities of LLMs: An Exploration in Mathematical Problem Solving. arXiv:2405.12205.
- Fatemi, M.; Rafiee, B.; Tang, M.; and Talamadupula, K. 2025. Concise Reasoning via Reinforcement Learning. arXiv:2504.05185.
- Gandhi, K.; Chakravarthy, A.; Singh, A.; Lile, N.; and Goodman, N. D. 2025. Cognitive Behaviors that Enable Self-Improving Reasoners, or, Four Habits of Highly Effective STaRs. arXiv:2503.01307.
- Gou, Z.; Shao, Z.; Gong, Y.; Shen, Y.; Yang, Y.; Duan, N.; and Chen, W. 2024. CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. arXiv:2305.11738.
- Griot, M.; Hemptinne, C.; Vanderdonckt, J.; and Yuksel, D. 2025. Large language models lack essential metacognition for reliable medical reasoning. *Nature communications*, 16(1): 642.
- Gui, L.; Gârbaacea, C.; and Veitch, V. 2024. BoNBon Alignment for Large Language Models and the Sweetness of Best-of-n Sampling. arXiv:2406.00832.
- Gulcehre, C.; Paine, T. L.; Srinivasan, S.; Konyushkova, K.; Weerts, L.; Sharma, A.; Siddhant, A.; Ahern, A.; Wang, M.; Gu, C.; Macherey, W.; Doucet, A.; Firat, O.; and de Freitas, N. 2023. Reinforced Self-Training (ReST) for Language Modeling. arXiv:2308.08998.
- Hao, S.; Gu, Y.; Ma, H.; Hong, J. J.; Wang, Z.; Wang, D. Z.; and Hu, Z. 2023. Reasoning with Language Model is Planning with World Model. arXiv:2305.14992.
- Havrilla, A.; Raparthy, S.; Nalmpantis, C.; Dwivedi-Yu, J.; Zhuravinskyi, M.; Hambro, E.; and Raileanu, R. 2024. GLoRe: When, Where, and How to Improve LLM Reasoning via Global and Local Refinements. arXiv:2402.10963.
- He, C.; Luo, R.; Bai, Y.; Hu, S.; Thai, Z. L.; Shen, J.; Hu, J.; Han, X.; Huang, Y.; Zhang, Y.; Liu, J.; Qi, L.; Liu, Z.; and Sun, M. 2024. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems. arXiv:2402.14008.
- Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. arXiv:2103.03874.
- Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; and Liu, T. 2025. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Transactions on Information Systems*, 43(2): 1–55.
- Ji, Y.; Li, J.; Xiang, Y.; Ye, H.; Wu, K.; Yao, K.; Xu, J.; Mo, L.; and Zhang, M. 2025. A Survey of Test-Time Compute: From Intuitive Inference to Deliberate Reasoning. arXiv:2501.02497.
- Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling Laws for Neural Language Models. arXiv:2001.08361.
- Lewkowycz, A.; Andreassen, A.; Dohan, D.; Dyer, E.; Michalewski, H.; Ramasesh, V.; Slone, A.; Anil, C.; Schlag, I.; Gutman-Solo, T.; Wu, Y.; Neyshabur, B.; Gur-Ari, G.; and Misra, V. 2022. Solving Quantitative Reasoning Problems with Language Models. arXiv:2206.14858.
- Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let’s Verify Step by Step. arXiv:2305.20050.
- Lin, S.; Hilton, J.; and Evans, O. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. arXiv:2109.07958.
- Luo, Y.; Song, Y.; Zhang, X.; Liu, J.; Wang, W.; Chen, G.; Su, W.; and Zheng, B. 2025. Deconstructing Long Chain-of-Thought: A Structured Reasoning Optimization Framework for Long CoT Distillation. arXiv:2503.16385.
- Ma, W.; He, J.; Snell, C.; Griggs, T.; Min, S.; and Zaharia, M. 2025. Reasoning Models Can Be Effective Without Thinking. *arXiv preprint arXiv:2504.09858*.
- Manuvinakurike, R.; Moss, E.; Watkins, E. A.; Sahay, S.; Raffa, G.; and Nachman, L. 2025. Thoughts without Thinking: Reconsidering the Explanatory Value of Chain-of-Thought Reasoning in LLMs through Agentic Pipelines. arXiv:2505.00875.

- Mirzadeh, I.; Alizadeh, K.; Shahrokhi, H.; Tuzel, O.; Bengio, S.; and Farajtabar, M. 2024. GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models. *arXiv:2410.05229*.
- Muennighoff, N.; Yang, Z.; Shi, W.; Li, X. L.; Fei-Fei, L.; Hajishirzi, H.; Zettlemoyer, L.; Liang, P.; Candès, E.; and Hashimoto, T. 2025. s1: Simple test-time scaling. *arXiv:2501.19393*.
- OpenAI. 2024. Learning to Reason with LLMs.
- Paul, D.; Ismayilzada, M.; Peyrard, M.; Borges, B.; Bosselut, A.; West, R.; and Faltings, B. 2024. RE-FINER: Reasoning Feedback on Intermediate Representations. *arXiv:2304.01904*.
- Qi, Z.; Ma, M.; Xu, J.; Zhang, L. L.; Yang, F.; and Yang, M. 2024. Mutual Reasoning Makes Smaller LLMs Stronger Problem-Solvers. *arXiv:2408.06195*.
- Rein, D.; Hou, B. L.; Stickland, A. C.; Petty, J.; Pang, R. Y.; Dirani, J.; Michael, J.; and Bowman, S. R. 2023. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. *arXiv:2311.12022*.
- Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *arXiv:1707.06347*.
- Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y. K.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv:2402.03300*.
- Snell, C.; Lee, J.; Xu, K.; and Kumar, A. 2024. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. *arXiv:2408.03314*.
- Sun, Y.; Wang, X.; Liu, Z.; Miller, J.; Efros, A. A.; and Hardt, M. 2020. Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. *arXiv:1909.13231*.
- Sun, Z.; Wang, Q.; Wang, H.; Zhang, X.; and Xu, J. 2025. Detection and Mitigation of Hallucination in Large Reasoning Models: A Mechanistic Perspective. *arXiv:2505.12886*.
- Team, K.; Du, A.; Gao, B.; Xing, B.; Jiang, C.; Chen, C.; Li, C.; Xiao, C.; Du, C.; Liao, C.; et al. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.
- Wang, C.; Feng, Y.; Chen, D.; Chu, Z.; Krishna, R.; and Zhou, T. 2025a. Wait, We Don't Need to "Wait"! Removing Thinking Tokens Improves Reasoning Efficiency. *arXiv:2506.08343*.
- Wang, P.; Li, L.; Shao, Z.; Xu, R. X.; Dai, D.; Li, Y.; Chen, D.; Wu, Y.; and Sui, Z. 2024. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations. *arXiv:2312.08935*.
- Wang, S.; Yu, L.; Gao, C.; Zheng, C.; Liu, S.; Lu, R.; Dang, K.; Chen, X.; Yang, J.; Zhang, Z.; Liu, Y.; Yang, A.; Zhao, A.; Yue, Y.; Song, S.; Yu, B.; Huang, G.; and Lin, J. 2025b. Beyond the 80/20 Rule: High-Entropy Minority Tokens Drive Effective Reinforcement Learning for LLM Reasoning. *arXiv:2506.01939*.
- Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *arXiv:2203.11171*.
- Wang, Y.; and Zhao, Y. 2024. Metacognitive Prompting Improves Understanding in Large Language Models. *arXiv:2308.05342*.
- Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; and Zhou, D. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv:2201.11903*.
- Welleck, S.; Lu, X.; West, P.; Brahman, F.; Shen, T.; Khashabi, D.; and Choi, Y. 2022. Generating Sequences by Learning to Self-Correct. *arXiv:2211.00053*.
- Wu, Y.; Wang, Y.; Du, T.; Jegelka, S.; and Wang, Y. 2025. When More is Less: Understanding Chain-of-Thought Length in LLMs. *arXiv preprint arXiv:2502.07266*.
- Xu, F.; Hao, Q.; Zong, Z.; Wang, J.; Zhang, Y.; Wang, J.; Lan, X.; Gong, J.; Ouyang, T.; Meng, F.; Shao, C.; Yan, Y.; Yang, Q.; Song, Y.; Ren, S.; Hu, X.; Li, Y.; Feng, J.; Gao, C.; and Li, Y. 2025. Towards Large Reasoning Models: A Survey of Reinforced Reasoning with Large Language Models. *arXiv:2501.09686*.
- Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; Zheng, C.; Liu, D.; Zhou, F.; Huang, F.; Hu, F.; Ge, H.; Wei, H.; Lin, H.; Tang, J.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Zhou, J.; Lin, J.; Dang, K.; Bao, K.; Yang, K.; Yu, L.; Deng, L.; Li, M.; Xue, M.; Li, M.; Zhang, P.; Wang, P.; Zhu, Q.; Men, R.; Gao, R.; Liu, S.; Luo, S.; Li, T.; Tang, T.; Yin, W.; Ren, X.; Wang, X.; Zhang, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Wang, Z.; Cui, Z.; Zhang, Z.; Zhou, Z.; and Qiu, Z. 2025a. Qwen3 Technical Report. *arXiv:2505.09388*.
- Yang, C.; Si, Q.; Duan, Y.; Zhu, Z.; Zhu, C.; Lin, Z.; Cao, L.; and Wang, W. 2025b. Dynamic Early Exit in Reasoning Models. *arXiv:2504.15895*.
- Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T. L.; Cao, Y.; and Narasimhan, K. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *arXiv:2305.10601*.
- Zhang, D.; Zhou, S.; Hu, Z.; Yue, Y.; Dong, Y.; and Tang, J. 2024. ReST-MCTS\*: LLM Self-Training via Process Reward Guided Tree Search. *arXiv:2406.03816*.
- Zhang, Z.; Zheng, C.; Wu, Y.; Zhang, B.; Lin, R.; Yu, B.; Liu, D.; Zhou, J.; and Lin, J. 2025. The Lessons of Developing Process Reward Models in Mathematical Reasoning. *arXiv:2501.07301*.
- Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E. P.; Zhang, H.; Gonzalez, J. E.; and Stoica, I. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *arXiv:2306.05685*.
- Zhu, Y.; Zhang, G.; Xu, C.; Shen, H.; Chen, X.; Wu, G.; and Wang, L. 2024. Efficient Test-Time Prompt Tuning for Vision-Language Models. *arXiv:2408.05775*.