Long-Short Chain-of-Thought Mixture Supervised Fine-Tuning Eliciting Efficient Reasoning in Large Language Models

Anonymous ACL submission

Abstract

Recent advances in large language models have demonstrated that Supervised Fine-Tuning (SFT) with Chain-of-Thought (CoT) reasoning data distilled from large reasoning models (e.g., DeepSeek R1) can effectively transfer reasoning capabilities to non-reasoning models. However, models fine-tuned with this approach inherit the "overthinking" problem from teacher models, producing verbose and redundant reasoning chains during inference. To address this challenge, we propose **L**ong-**S**hort Chain-of-Thought **Mixture** **S**upervised **F**ine-**T**uning (**LS-Mixture SFT**), which combines the long CoT reasoning dataset with their short counterparts obtained through structure-preserved rewriting. Our experiments demonstrate that models trained with the LS-Mixture SFT method achieved an average accuracy improvement of 2.3% across various benchmarks compared to those trained with standard SFT. Furthermore, this approach substantially reduced the model response length by approximately 47.61%. This work offers an approach to endow non-reasoning models with reasoning capabilities through supervised fine-tuning while avoiding the inherent overthinking problems inherited from teacher models, thereby enabling efficient reasoning in the fine-tuned models.

1 Introduction

The emergence of large reasoning models (LRMs) (Chen et al., 2025a), such as DeepSeek R1 (DeepSeek-AI et al., 2025) and OpenAI o1 (OpenAI, 2024), have demonstrated remarkable reasoning abilities in complex tasks by generating explicit chain-of-thoughts (CoT) (Wei et al., 2022) closed by special tokens (`<think>` and `</think>`) about a question before arriving at the final answer. Recent works (Huang et al., 2024; Min et al., 2024) have shown that advanced reasoning abilities can be transferred from LRMs to non-reasoning large language models (LLMs) through supervised fine-tuning (SFT) on high-quality CoT reasoning data distilled from LRM (DeepSeek-AI et al., 2025; Muennighoff et al., 2025).

Existing open-source efforts, such as s1 (Muennighoff et al., 2025), Sky-T1 (Team, 2025a) and LIMO (Ye et al., 2025), have demonstrated that non-reasoning LLMs as student models can be effectively transformed into reasoning-capable models through supervised fine-tuning on long CoT trajectories distilled from LRMs as teacher models. Although training on distilled datasets successfully elicits reasoning abilities in foundation models, it also causes these models to inherit the inherent overthinking problem (Chen et al., 2025b) of the original LRM (Sui et al., 2025; Wang et al., 2025c). Several recent studies have sought to address the overthinking problem of LRM during training and inference from the perspectives of reinforcement learning and inference-time optimization, aiming to achieve efficient reasoning. However, there remains a lack of research on how to prevent student models from inheriting the overthinking issue of teacher models during the distillation stage. Thus, we propose the problem: "**how can data distillation and supervised fine-tuning be leveraged to elicit more *efficient* reasoning abilities in non-reasoning models**—specifically, enabling them to avoid inheriting the overthinking problem from teacher models?".

In this paper, we propose a novel solution to this problem: Long-Short Chain-of-Thought Mixture Supervised Fine-Tuning (**LS-Mixture SFT**). Our approach first performs structure-preserved rewriting of the reasoning paths in the dataset with long CoT trajectories distilled from LRM, resulting in a corresponding dataset with short CoT reasoning paths. We then construct a mixture of both long and short CoT reasoning datasets, and use it to perform supervised fine-tuning on the student model. This mixture allows student models to learn both com-

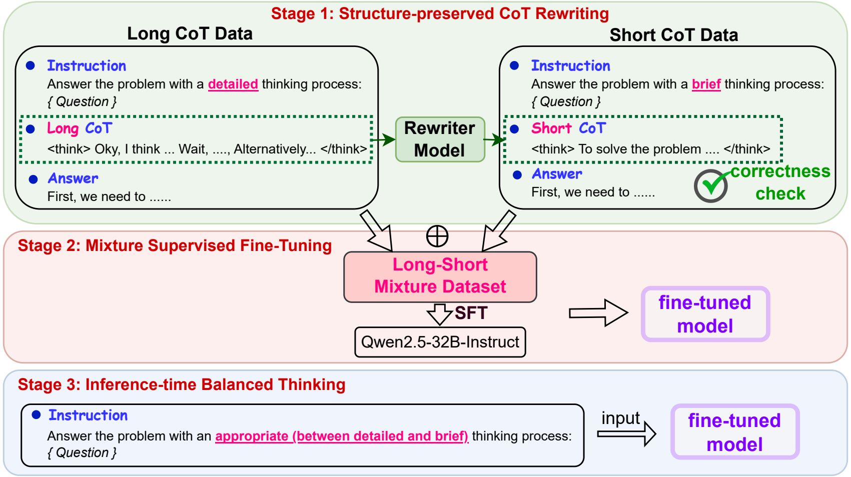

Stage 1: Structure-preserved CoT Rewriting
Long CoT Data
Short CoT Data
Instruction
Answer the problem with a detailed thinking process:
{ Question }
Long CoT
<think> Oky, I think ... Wait, ..., Alternatively... </think>
Answer
First, we need to ......
Rewriter Model
Instruction
Answer the problem with a brief thinking process:
{ Question }
Short CoT
<think> To solve the problem .... </think>
Answer
First, we need to ......
correctness check
Stage 2: Mixture Supervised Fine-Tuning
⊕
Long-Short Mixture Dataset
SFT
Qwen2.5-32B-Instruct
fine-tuned model
Stage 3: Inference-time Balanced Thinking
Instruction
Answer the problem with an appropriate (between detailed and brief) thinking process:
{ Question }
input
fine-tuned model

Figure 1: Overview of **LS-Mixture SFT**. This method consists of three stages: **1) Structure-preserved CoT Rewriting**: A LLM is used to rewrite the long CoT trajectories into short ones while preserving the core structure. **2) Mixture Supervised Fine-Tuning**: Non-reasoning LLM is been supervised fine-tuned on mixture datasets. **3) Inference-time Balanced Thinking**: The fine-tuned model is designed to employ a balanced thinking mode that lies between detailed and brief when generating reasoning responses to queries.

prehensive reasoning patterns and efficient reasoning shortcuts, resulting in models that can generate more efficient reasoning during inference without sacrificing accuracy. Our approach can directly reuse existing long CoT reasoning datasets without incurring the substantial costs associated with additional data distillation. Specifically, we created a mixed dataset of long and short reasoning chains, **s1K-mix**, based on the existing s1K-1.1 dataset (Muennighoff et al., 2025), and utilized this mixed dataset to train the Qwen2.5-32B-Instruct model, resulting in our model **s1-mix-32B**.

Our extensive experiments across three challenging reasoning benchmarks validate the effectiveness of the LS-Mixture SFT approach. The experimental results demonstrate that our s1-mix-32B model achieves higher accuracy on MATH500, AIME24, and GPQA benchmarks (improvements of 2.2%, 6.7%, and 2%, respectively) compared to models trained solely on long-chain reasoning data, while significantly reducing average response length (by 47.61% on average). Ablation studies further confirm the importance of our proposed structure-preserved CoT rewriting strategy and the advantages of the long-short chain mixture training method in balancing reasoning efficiency and accuracy. These findings indicate that LS-Mixture SFT not only effectively elicits reasoning capabilities in non-reasoning models but also successfully avoids the overthinking problem inherited from existing LRMs, providing an effective approach for training more efficient reasoning models.

Our contributions can be summarized as follows:

- We propose a novel method for transforming long chain-of-thought trajectories into their short counterparts: Structure-preserved CoT Rewriting, which is designed to rewrite reasoning paths while preserving the core structure, thereby preventing overly liberal rewriting that might cause models to lose crucial "aha moments" ability during training.

- We introduce LS-Mixture SFT, a novel fine-tuning approach that mix long and short reasoning dataset to elicits efficient reasoning in large language models.

- Based on these methods, we build a new mixture dataset **s1K-mix** and a fine-tuned model **s1-mix-32B** released on HuggingFace.

- Through extensive experiments, we demonstrate that our approach significantly reduces model response length during inference while improving task performance.

- During our experiments, we observed an interesting phenomenon: the fine-tuned model's ability to success in balanced thinking was not explicitly trained but rather emerged as a natural consequence of training on a mixture dataset comprising both long-chain and short-chain reasoning examples.

Our code, model, and dataset are open-sourced at GitHub and HuggingFace.

2 Methodology

In this section, we introduce Long-Short Chain-of-Thought Mixture Supervised Fine-Tuning (**LS-Mixture SFT**), our novel approach for efficiently transferring reasoning capabilities from LRMs to non-reasoning LLMs. The key insight is that not all reasoning steps contribute equally to the final solution—many tokens in verbose CoT are redundant. By leveraging LLMs as rewriter model that preserve the core reasoning structure while eliminating redundancy, we create a complementary dataset of short CoT. These shortened trajectories maintain the core structure and key steps necessary for accurate problem-solving but with significantly reduced token counts. When mixed with the original long CoT dataset, the combination allows student models to learn both comprehensive reasoning patterns and efficient reasoning shortcuts, resulting in model that can generate more concise CoT during inference without sacrificing accuracy.

As illustrated in Figure 1, our approach can be divided into three stages: **(1) Structure-preserved CoT Rewriting**: We use a LLM as a rewriter model for the structure-preserved rewriting of long CoT. This process incorporates specific constraints in the instruction prompt to ensure that the rewriting process maintains the core logical structure and keys steps of the original CoT. Based on the existing long CoT dataset, this stage produces a corresponding short CoT dataset. **(2) Mixture Supervised Fine-Tuning**: The original long CoT reasoning dataset and short CoT reasoning dataset obtained in the previous stage are completely randomly mixed to create a long-short mixture dataset. This mixed dataset is then used to perform supervised fine-tuning on a non-reasoning LLM. **(3) Inference-time Balanced Thinking**: The mixture of both long and short CoT datasets enables student models to acquire comprehensive reasoning patterns while simultaneously learning efficient reasoning shortcuts. During inference, our model is provided with instructions of balanced thinking mode to solve the problem.

In the following subsections, we first provide a formal definition of the task (2.1), followed by a detailed explanation of each stage of our method (2.2, 2.3, 2.4).

2.1 Formal Task Definition

Let $D_{\text{long}} = \{(x_i, r_i^L, y_i)\}_{i=1}^N$ denote a long CoT dataset comprising N instances, where x_i represents a complex question, r_i corresponds to the long CoT trajectory distilled from a LRM, and y_i denotes the corresponding answer.

Our objective is to utilize this dataset through SFT to endow a non-reasoning LLM with effective reasoning capabilities.

2.2 Structure-preserved CoT Rewriting

A key component of our LS-Mixture SFT approach is the **structure-preserved CoT rewriting** method, which transforms verbose long CoT into more concise versions while preserving their core logical structure and key reasoning steps. This method significantly shortens the thinking part in the training data while preserving the reasoning process demonstrated by LRMs when addressing a problem, particularly the "aha moments" phenomenon exhibited by these reasoning-capable models.

We employ another LLM (Qwen2.5-72B-Instruct) as the rewriter model $\mathbb{P}_{\text{rewriter}}$, incorporating explicit constraints in the prompt template to instruct the model to maintain the original logical structure and key steps of the CoT trajectory during rewriting. The prompt template employed by the rewriter model is presented in Appendix E.1. Additionally, Appendix I presents a case study of Chain-of-Thought (CoT) rewriting.

For each data point in the dataset D_{long}, we utilize the rewriter model to transform the long CoT trajectory r_i^L into a shorter one r_i^S, which can be formally expressed as:

$$r_i^S = \mathbb{P}_{\text{rewriter}}(r_i^L | x_i) \tag{1}$$

After structure-preserved CoT rewriting, the short CoT are substantially shorter in length compared to their long CoT counterparts. Utilizing these rewritten short CoT trajectories, we are able to construct a short CoT dataset $D_{\text{short}} = \{(x_i, r_i^S, y_i)\}_{i=1}^N$.

2.3 Mixture Supervised Fine-Tuning

Following the previous stage that yields the short reasoning dataset D_{short}, we proceed to completely randomly merge it with the original long reasoning dataset D_{long}, creating a new mixed dataset D_{mix}:

$$D_{\text{mix}} = D_{\text{long}} \cup D_{\text{short}} \tag{2}$$

This mixture dataset D_{mix} is then utilized to perform SFT on a non-reasoning LLM M aiming to eliciting its efficient reasoning. To align with the current output format of LRMs, we encapsulate the

CoT trajectory using special tokens `<think>` and `</think>`, and concatenate it with the answer part to form the ground-truth response for fine-tuning. The optimization objective M^* can be formulated as follows:

$$L(D_{\text{long}}) = \sum_{D_{\text{long}}} -\log \mathbb{P}_M(r_i^L \oplus y_i | x_i, p_{\text{L}}) \quad (3)$$

$$L(D_{\text{short}}) = \sum_{D_{\text{short}}} -\log \mathbb{P}_M(r_i^S \oplus y_i | x_i, p_{\text{S}}) \quad (4)$$

$$M^* = \arg\min_M L(D_{\text{long}}) + L(D_{\text{short}}) \quad (5)$$

In Equations 3 and 4, p_{L} and p_{S} respectively represent the prompt that instruct the language model to reasoning in detailed and brief thinking modes. The specific prompt templates can be found in Appendix E.2 and E.3.

The mixture dataset ensures that the model is exposed to both comprehensive thinking patterns from long CoT trajectories and the efficient patterns from short ones, which enables the model to adapt its reasoning pattern based on the instruction type. When prompted with "detailed thinking" instructions, the model demonstrates comprehensive reasoning inherited from long CoT examples. Simultaneously, under "brief thinking" instructions, it employs concise yet effective reasoning patterns learned from short CoT examples.

2.4 Inference-time Balanced Thinking

Through our mixture training approach, the model simultaneously acquires both detailed and concise thinking modes. However, neither mode achieves an optimal balance between response effectiveness and efficiency. To address this limitation, we propose an inference-time balanced thinking methodology that leverages the dual reasoning capabilities developed during training while optimizing for both effectiveness and efficiency during model deployment.

To implement **balanced thinking mode**, we maintain the format of prompt template between the inference time and the training time, while modifying the instructions regarding the thinking mode. Specifically, we replace the directives for either detailed or brief thinking with instructions that encourage the model to engage in an "appropriate" thinking process that falls between these two extremes. This approach enables the model to balance effectiveness and efficiency in its reasoning process. The formulation can be expressed as follows:

$$(r_i, y_i) = \mathbb{P}_{M^*}(x_i | p_{\text{B}}) \quad (6)$$

where r_i is the approximate reasoning chain that is generated by the post-trained model M^*, and p_{B} is the prompt template for balanced thinking. The specific prompt template can be found in Appendix E.4.

3 Experiments

3.1 Dataset

To demonstrate the effectiveness of the LS-Mixture SFT method, we conducted experimental evaluations based on two open-source datasets: s1K-1.1 and OpenThoughts-2K.

s1K-1.1 and s1K-mix Dataset **s1K-1.1** (Muennighoff et al., 2025) contains 1,000 instances of detailed reasoning trajectories and answers distilled from the DeepSeek-R1 model. We implemented our structure-preserved CoT rewriting technique using Qwen2.5-72B-Instruct as the rewriter model. During rewriting process, 16 instances exceeded context length limitations, resulting in their exclusion from the dataset. The final short reasoning chain dataset (D_{short}) consisted of 984 examples. The mixture of these long and short examples constitutes our **s1K-mix** dataset. The statistics for these datasets are presented in Appendix F. For the purpose of clarity and to adhere to the naming conventions established in prior research, we designate the model trained on s1K-1.1 as s1-32B and the model trained on s1K-mix as s1-mix-32B.

OpenThoughts-2K The OpenThoughts dataset (Guha et al., 2025) comprises 114K high-quality synthetic reasoning data samples, covering multiple domains including mathematics, science, and programming. Due to computational resource constraints, we randomly sampled 2K samples from the 114K instances to form a subset, which constitutes the **OpenThoughts-2K** dataset employed in this study. Following a similar methodology to that employed in constructing the s1K-mix dataset, we developed the **OpenThoughts-2K-mix** dataset based on OpenThoughts-2K utilizing the mixture strategy proposed by our method.

Table 1: Results on 3 benchmarks. For each benchmark, we report both the response accuracy and response length in our evaluation results (with the exception of the o1 model). Due to accessibility limitations of the o1 model, we only report their publicly available scores on these benchmarks. Among these baseline models, s1.1-32B serves as our primary baseline model for comparison.

Model/Method	**Dataset Size**	**SFT or RL**	**MATH500**		**AIME24**		**GPQA**		**Avg. Length**
			Acc	Length	Acc	Length	Acc	Length	
Baselines of API only									
o1-previwe	unknown	N/A	85.5	N/A	44.6	N/A	73.3	N/A	N/A
o1-mini	unknown	N/A	90	N/A	70	N/A	60	N/A	N/A
o1	unknown	N/A	94.8	N/A	74.4	N/A	77.3	N/A	N/A
DeepSeek-R1$_{(671B)}$	unknown	N/A	96.8	7,658.1	73.3	27,090.2	75.7	23,696.2	12,820.9
Baselines of Open Weights									
R1-Distill-Qwen-32B	unknown	SFT	93.8	8,044.4	60.0	24,786.1	61.7	25,135.9	13,382.8
Sky-T1-32B	17K	SFT	85	6,839.1	50.0	7,893.9	53	10,376.5	7,844.6
Sky-T1-32B-Flash	10k	RL	84.2	3,873.7	26.7	14,037.5	51.5	4,428.6	4,443.5
LIMO$_{(32B)}$	817	SFT	93.8	10,352.7	53.3	46,604.6	59.6	23,635.1	15,459.1
O1-Pruner$_{(32B)}$	5K	RL	90.6	3,784.0	33.3	11,390.7	44.4	6,434.4	4,818.3
SimpleRL-Zoo$_{(32B)}$	24K	RL	82.4	1,756.3	16.7	3,272.9	44.4	1,588.2	1,773.1
CoT-Valve$_{(32B)}$	8K	SFT	88.8	11,584.4	43.3	47,319.8	54.5	47,970.5	22,953.2
Qwen2.5-32B-Instruct + s1K-1.1									
SFT$^{☆}$	1K	SFT	92.4$_{\pm 1.6}$	12,351.4	53.3$_{\pm 6.7}$	53,455.6	59.1$_{\pm 2.0}$	56,040.7	25,927.8
LS-Mixture SFT $^{★}$	1.98K	SFT	**94.6**$_{\pm 2.0}$	**8,648.7**	**60.0**$_{\pm 6.7}$	**40,251.3**	**61.1**$_{\pm 2.5}$	**21,995.7**	**13,581.1**$_{\downarrow 47.6\%}$
Qwen2.5-32B-Instruct + OpenThoughts-2K									
SFT	2K	SFT	91.7$_{\pm 0.7}$	13,604.1	53.3$_{\pm 3.3}$	55,399.3	58.1$_{\pm 2.5}$	53,110.0	26,071.2
LS-Mixture SFT	4K	SFT	**94.4**$_{\pm 0.6}$	**6,978.2**	**63.3**$_{\pm 3.3}$	**31,356.8**	**61.6**$_{\pm 2.0}$	**22,545.8**	**12,216.9**$_{\downarrow 53.1\%}$

☆ also referred to as the **s1.1-32B** model.
★ also referred to as the **s1-mix-32B** model.

3.2 Experiment Setup

Training We perform supervised fine-tuning on Qwen2.5-32B-Instruct using the dataset **s1K-mix** and OpenThoughts-2K-mix using basic hyper parameters outline in Appendix G. All model training was conducted using the LlamaFactory (Zheng et al., 2024). The relevant training hyper parameters are maintained consistent with those used for the **s1.1-32B** model (Muennighoff et al., 2025). Given that the mixture dataset contains a greater number of samples than raw dataset, we adjusted the number of epochs to ensure that both models were exposed to an equivalent quantity of training samples. Taking the experiment using the s1K-1.1 dataset as an example: let N_{long} denote the number of training epochs for s1.1-32B ($N_{\text{long}} = 5$), N_{mix} represent the number of training epochs for our s1-mix-32B model, and $|D_{\text{long}}|$ and $|D_{\text{mix}}|$ denote the size of the respective datasets. The numerical relationship is represented as: $N_{\text{long}} \times |D_{\text{long}}| = N_{\text{mix}} \times |D_{\text{mix}}|$.

Baselines The experimental comparisons involve three categories of baseline methods: **(1) Baselines of API only**. These models are all commercial closed-source models, including OpenAI o1-series (OpenAI, 2024) and DeepSeek-R1 (DeepSeek-AI et al., 2025). **(2) Baselines of Open Weights**. These baseline models were trained using diverse methods and have publicly released their weights, including R1-Distill-Qwen-32B (DeepSeek-AI et al., 2025), Sky-T1-32B (Team, 2025a), Sky-T1-32B-Flash (Team, 2025b), LIMO (Ye et al., 2025), O1-Pruner (Luo et al., 2025), SimpleRL-Zoo (Zeng et al., 2025), and CoT-Valve (Ma et al., 2025). These models employ either supervised fine-tuning (SFT) or reinforcement learning (RL) approaches, with their specific correspondences systematically summarized in Table 1. Among them, the O1-Pruner baseline selected the version based on QwQ-32B-Preview. **(3) Standard SFT**. Standard SFT refers to the approach that utilizes only the original distilled dataset for model training. In contrast, LS-Mixture SFT incorporates a compressed short CoT dataset during the SFT process.

Benchmarks We evaluate the models on five in-domain benchmarks and two out-of-distribution benchmarks to evaluate their performance. **(1) In-domain Evaluation**: the American Invitational Mathematics Examination (**AIME24** and

AIME25), **MATH500** (Hendrycks et al., 2021), the American Mathematic Competitions (**AMC23**) and **GPQA Diamond** which consists of 198 PhD-level science questions from Biology, Chemistry, and Physics. The in-domain evaluation results are presented in Table 1 and Appendix A. **(2) Out-of-distribution Evaluation**: the 1,000 crowd-sourced Python programming problems (**MBPP**) and the 164 programming problems released by OpenAI (**HumanEval**). The out-of-distribution evaluation results are presented in Appendix B. More specific evaluation details are provided in Appendix D.

Response Length Evaluation In addition to evaluating accuracy on the benchmarks, we computed the average response character length generated by models. This metric is crucial for assessing inference efficiency, as shorter responses directly translate to reduced latency and computational costs. We operate under the principle that, given comparable levels of accuracy, models that produce shorter responses are inherently more efficient and practical for real-world applications. For each model, we calculated the weighted average lengths across all benchmarks, using the number of samples in each evaluation dataset as weights for computation.

3.3 Results

Table 1 presents the experimental results on the three benchmarks, highlighting the key findings: **our proposed method LS-Mixture achieves a substantial reduction in model response length while imporving answer accuracy.** Despite utilizing the same training question set and equivalent number of training instances as s1.1-32B, our **s1-mix-32B** model attains accuracy improvements of 2.2% on MATH500 (from 92.4% to 94.6%), 6.7% on AIME24 (from 53.3% to 60%), and 2% on GPQA (from 59.1% to 61.1%), all while reducing average response length by 47.61% compared to s1.1-32B. Similar trends were also observed in the evaluation results when using OpenThoughts-2K as the training dataset. These results demonstrate the effectiveness of our proposed method in enhancing both reasoning accuracy and efficiency.

4 Ablations

4.1 Impact of Rewriting Strategies

To investigate the importance of our CoT rewriting strategy, we performed ablation studies to compare our approach with two alternative rewriting strategies: **Direct compression**: A straightforward approach where the LLM is instructed to compress the long reasoning chain freely. The specific prompt templates can be found in Appendix E.5. **ThinkTwice**: this approach (Tian et al., 2025) incorporates the answer into the specific prompt used to model generation. The thinking part produced during generation serve as the shortened CoT.

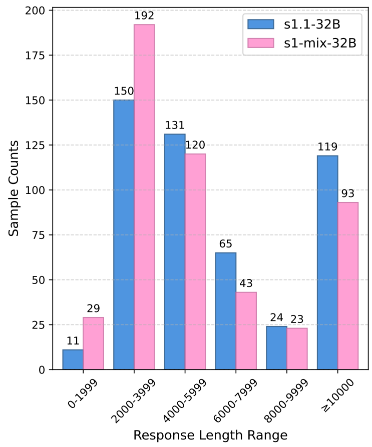


Figure 2: Comparison of response length distribution between **s1.1-32B** and **s1-mix-32B** models on the MATH500 benchmark. The horizontal axis represents response length, while the vertical axis shows the number of samples within each length range.

For each rewriting strategy, we created a corresponding short reasoning dataset and applied our method to fine-tune Qwen2.5-32B-Instruct. As demonstrated in the Table 2, all alternative chain-of-thought rewriting methods resulted in diminished model training effectiveness, highlighting the importance of preserving the original logical reasoning structure during rewriting stage.

4.2 Impact of Long-Short Mixing Strategies

To investigate the effectiveness of our proposed mixing strategy, We performed ablation studies to compare our approach with two alternative mixing strategies: **Long-only**: The thinking part of the data point exclusively comprises long CoT trajectories, specifically D_{long}. **Short-only**: The thinking part exclusively comprises short CoT trajectories, specifically D_{short}.

Table 3 presents the experimental results across our evaluation benchmarks. The results demonstrate that our proposed mixing strategy consistently outperforms other approaches.

4.3 Impact of Inference-time Thinking Modes

During the training of **s1-mix-32B**, detailed and brief thinking modes were employed for the long

Table 2: Ablation experiment results on CoT rewriting strategies. We employed direct rewriting strategy and the ThinkTwice method to obtain short CoT trajectories.

Strategy	**MATH500**		**AIME24**		**GPQA**	
	Acc.	Length	Acc.	Length	Acc.	Length
Direct	85.4	6,106.1	33.3	5,876.4	53.5	31,204.0
ThinkTwice	91	13,027.8	58.1	47,520.3	43.3	47,928.5
Structure-preserved	**94.6**	8,648.7	**60**	40,251.3	**61.1**	21,995.7

Table 3: Ablation experiment results on dataset mixing strategies. We experimented with three datasets created by different strategies: only the long CoT reasoning dataset, only the short CoT reasoning dataset, and mixture dataset.

Mix Method	**MATH500**		**AIME24**		**GPQA**	
	Acc.	Length	Acc.	Length	Acc.	Length
Long-only	92.4	12,351.4	53.3	53,455.6	59.1	56,040.7
Short-only	82.6	3,205.8	16.7	6,646.3	49.0	3,961.7
Mixture	**94.6**	8,648.7	**60**	40,251.3	**61.1**	21,995.7

Table 4: Ablation experiment results on different thinking modes. We evaluated the performance of the **s1-mix-32B** model using three thinking modes: detailed thinking, brief thinking, and balanced thinking.

Thinking Mode	**MATH500**		**AIME24**		**GPQA**	
	Acc.	Length	Acc.	Length	Acc.	Length
Brief	81.0	2,963.9	20.0	4,490.3	52.0	4,125.0
Detailed	92.6	1,162.3	56.7	53,107.4	**62.1**	41,762.9
Balanced	**94.6**	8,648.7	**60**	40,251.3	61.1	21,995.7

and short CoT dataset, while a balanced thinking mode was utilized during inference. To investigate the impact of different thinking modes at inference time, we conducted evaluations using these three thinking modes for **s1-mix-32B**. As shown in Table 4, employing the balanced thinking mode yields the optimal results, which validate our hypothesis that the balanced thinking mode during inference time can effectively leverage both the comprehensive thinking capabilities learned from long CoT examples and the efficient reasoning patterns acquired from short counterparts.

5 Discussion

Structure-preserving in CoT Rewriting. Our findings highlight the importance of maintaining the core logical structure when rewriting long CoT into short formats. Our ablation studies 4.1, which explore various strategies for CoT trajectory rewriting, revealed a insight: overly simplified CoT fail to adequately stimulate the student model's reasoning capabilities. Conversely, we observed that by preserving the original core structure and key steps from the long CoT during the rewriting stage could the student model be guided to learn how to reason effectively.

Exploration of Mixing Ratios. Our previous experiments assumed a mixing ratio of $1:1$ between long and short reasoning chains. To investigate the effects of mixing ratios on the final results, we modified the mixing ratio to $1:\alpha$ and conducted experiments with α values of 0, 0.25, 0.5, 0.75, and 1, respectively. Figure 3 presents the experimental results on the s1K-1.1 dataset and Qwen2.5-7B-Instruct model under different mixing ratios.

Relationship between Accuracy and Response Length. In the large reasoning model, increased response length often signifies the emergence of the "aha moment" phenomenon and correlates with improved model performance. However, recent studies (Wang et al., 2025a; Ghosal et al., 2025) reveal that excessive deliberation-induced lengthening can degrade performance, suggesting that mitigating overthinking appropriately enhances both model accuracy and response conciseness. These findings align with our experimental results.

6 Related Work

6.1 Chain-of-Thought Reasoning in LLMs

Chain-of-Thought reasoning has emerged as a pivotal technique for enhancing the reasoning capa-

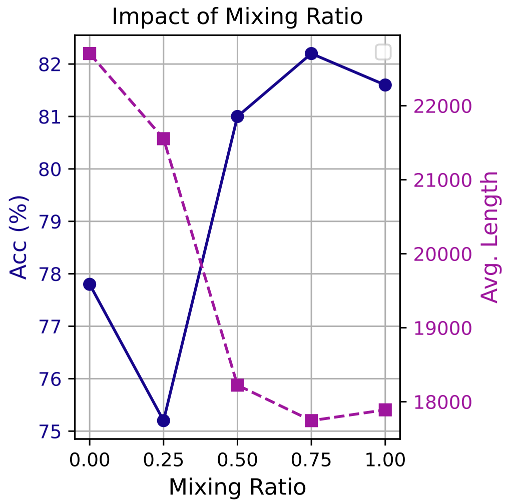


Figure 3: The impact of mixing ratios α. The x-axis represents the mixing ratio, while the y-axis displays the model's accuracy and average response length on the benchmark after training.

bilities of large language models. Initially introduced by Wei et al. (2022), CoT prompting encourages models to generate intermediate reasoning steps before producing a final answer (Byun et al., 2024). This approach has proven particularly effectively for complex reasoning tasks (Li et al., 2024; Madaan et al., 2023), including mathematical problem-sovling (Yin et al., 2024), logical reasoning (Wan et al., 2024; Toroghi et al., 2024), and scientific inquery (Sun et al., 2024).

With the recent discovery of test-time scaling laws (Wu et al., 2024; Snell et al., 2025), Large Reasoning Models, exemplified by DeepSeek R1 (DeepSeek-AI et al., 2025), have undergone substantial development. These works utilize techniques such as reinforcement learning to enable LLMs to generate a CoT reasoning process enclosed by special tokens (e.g., `<think>` and `</think>`) (Qin et al., 2024; Team et al., 2025; Wen et al., 2025).

The emergence of LRMs has further enhanced the capabilities of LLM on complex reasoning tasks. However, to transfer these reasoning abilities to non-reasoning models such as Qwen-2.5 series (Qwen et al., 2025), current research (Zhang et al., 2025a; Muennighoff et al., 2025) has found that it is also possible to elicit thier reasoning abilities by performing supervised fine-tuning on non-reasoning models using dataset distilled from LRMs (Zhang et al., 2025b; Chen et al., 2025c). Our approach follows this line of research, refining the fine-tuning methodology to elicit efficient reasoning capabilities in non-reasoning models.

6.2 Efficient Reasoning in LRMs

While LRMs improve performance in System-2 reasoning domains (Li et al., 2025), they also introduce significant computational overheads due to verbose and redundant reasoning steps, known as the "overthinking phenomenon" (Qu et al., 2025; Sui et al., 2025). To address this issue, a series of efficient reasoning (Feng et al., 2025; Xu et al., 2025; Cui et al., 2025; Liu et al., 2025; Yang et al.) methods have been proposed to enhance the inference-time efficiency of LRMs. These methods vary in approach: some incorporate response length-related rewards into reinforcement learning (Aggarwal and Welleck, 2025; Luo et al., 2025; Shen et al., 2025; Yeo et al., 2025; Zeng et al., 2025), others differentiate problem difficulty levels to allocate token budgets accordingly (Ong et al., 2025; Aytes et al., 2025; Huang et al., 2025), and yet others leverage smaller models to achieve faster thinking processes (Akhauri et al., 2025; Wang et al., 2025b). To our knowledge, our approach inspired by C3oT (Kang et al., 2025) is the first investigation from the supervised fine-tuning perspective on achieving efficient reasoning goals while eliciting reasoning capabilities in non-reasoning models through distillation from LRMs.

7 Conclusion

We presented **LS-Mixture SFT**, a novel approach for eliciting efficient reasoning capabilities in non-reasoning models using dataset distilled from large reasoning models, thereby enabling the trained model to maintain task performance while reducing response length, and avoiding the inheritance of the overthinking problem from teacher models to student models during the distillation process. We found that our approach of mixing long and short reasoning chains can effectively enhance the performance of reasoning distillation and can be applied to SFT in various scenarios. This method is straightforward and enables supervised fine-tuned models to effectively reduce the length of reasoning chains while maintaining accuracy.

In future work, we aim to investigate the integration of our supervised fine-tuning approach with current reinforcement learning methods and token-level compression techniques to further optimize efficient reasoning of model.

Limitations

Despite the promising results presented in this paper, our study is subject to several limitations. The experiments conducted in this work were restricted to a 32B parameter model and datasets containing only 1,000 examples (s1K-1.1 dataset) and 2,000 examples (OpenThoughts-2K dataset). Due to computational resource constraints, we were unable to extend our experiments to larger-scale models or more extensive datasets.

Furthermore, since the mixing ratio is a continuous variable, experimental identification of the optimal ratio necessitates densely sampled design exploration. Constrained by computational resources, this study evaluates only a set of representative mixing ratios for the long-short CoT mixture. The optimal balance between these different types of reasoning demonstrations may vary across different model sizes, tasks, and domains. This represents an important dimension for future exploration that could yield further improvements in model performance and efficiency.

Ethics Statement

This research utilizes the s1K-1.1 dataset (Muennighoff et al., 2025), OpenThoughts dataset (Guha et al., 2025) and Qwen2.5 series models (Qwen et al., 2025), both of which are publicly available online resources. We have provided appropriate citations to acknowledge the original work behind these resources. Our study focuses on improving model training and inference efficiency through Chain-of-Thought trajectory rewriting techniques, which does not introduce new ethical concerns beyond those inherent to large language model research.

References

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *Preprint*, arXiv:2503.04697.

Yash Akhauri, Anthony Fei, Chi-Chih Chang, Ahmed F. AbouElhamayed, Yueying Li, and Mohamed S. Abdelfattah. 2025. Splitreason: Learning to offload reasoning. *Preprint*, arXiv:2504.16379.

Simon A. Aytes, Jinheon Baek, and Sung Ju Hwang. 2025. Sketch-of-thought: Efficient llm reasoning with adaptive cognitive-inspired sketching. *Preprint*, arXiv:2503.05179.

Ju-Seung Byun, Jiyun Chun, Jihyung Kil, and Andrew Perrault. 2024. ARES: Alternating reinforcement learning and supervised fine-tuning for enhanced multi-modal chain-of-thought reasoning through diverse AI feedback. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4410–4430, Miami, Florida, USA. Association for Computational Linguistics.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025a. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *Preprint*, arXiv:2503.09567.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025b. Do not think that much for 2+3=? on the overthinking of o1-like llms. *Preprint*, arXiv:2412.21187.

Zhipeng Chen, Yingqian Min, Beichen Zhang, Jie Chen, Jinhao Jiang, Daixuan Cheng, Wayne Xin Zhao, Zheng Liu, Xu Miao, Yang Lu, Lei Fang, Zhongyuan Wang, and Ji-Rong Wen. 2025c. An empirical study on eliciting and improving r1-like reasoning models. *Preprint*, arXiv:2503.04548.

Yingqian Cui, Pengfei He, Jingying Zeng, Hui Liu, Xianfeng Tang, Zhenwei Dai, Yan Han, Chen Luo, Jing Huang, Zhen Li, Suhang Wang, Yue Xing, Jiliang Tang, and Qi He. 2025. Stepwise perplexity-guided refinement for efficient chain-of-thought reasoning in large language models. *Preprint*, arXiv:2502.13260.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. Efficient reasoning models: A survey. *Preprint*, arXiv:2504.10903.

Clémentine Fourrier, Nathan Habib, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. 2023. Lighteval: A lightweight framework for llm evaluation.

Soumya Suvra Ghosal, Souradip Chakraborty, Avinash Reddy, Yifu Lu, Mengdi Wang, Dinesh Manocha, Furong Huang, Mohammad Ghavamzadeh, and Amrit Singh Bedi. 2025. Does thinking more always help? understanding test-time scaling in reasoning models. *Preprint*, arXiv:2506.04210.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, and 31 others. 2025. Openthoughts: Data recipes for reasoning models. *Preprint*, arXiv:2506.04178.

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. *Preprint*, arXiv:2103.03874.

Chengsong Huang, Langlin Huang, Jixuan Leng, Jiacheng Liu, and Jiaxin Huang. 2025. Efficient test-time scaling via self-calibration. *Preprint*, arXiv:2503.00031.

Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. 2024. O1 replication journey – part 2: Surpassing o1-preview through simple distillation, big progress or bitter lesson? *Preprint*, arXiv:2411.16489.

HuggingFace. 2025. Open r1: A fully open reproduction of deepseek-r1.

Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. 2025. C3ot: Generating shorter chain-of-thought without compromising effectiveness. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24312–24320.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. 2024. Chain of thought empowers transformers to solve inherently serial problems. In *The Twelfth International Conference on Learning Representations*.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhiwei Li, Bao-Long Bi, Ling-Rui Mei, Junfeng Fang, Zhijiang Guo, Le Song, and Cheng-Lin Liu. 2025. From system 1 to system 2: A survey of reasoning large language models. *Preprint*, arXiv:2502.17419.

Yule Liu, Jingyi Zheng, Zhen Sun, Zifan Peng, Wenhan Dong, Zeyang Sha, Shiwen Cui, Weiqiang Wang, and Xinlei He. 2025. Thought manipulation: External thought can be efficient for large reasoning models. *Preprint*, arXiv:2504.13626.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *Preprint*, arXiv:2501.12570.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025. Cot-valve: Length-compressible chain-of-thought tuning. *Preprint*, arXiv:2502.09601.

Aman Madaan, Katherine Hermann, and Amir Yazdanbakhsh. 2023. What makes chain-of-thought prompting effective? a counterfactual study. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1448–1535, Singapore. Association for Computational Linguistics.

Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, Wayne Xin Zhao, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. 2024. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *Preprint*, arXiv:2412.09413.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Preprint*, arXiv:2501.19393.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. 2025. RouteLLM: Learning to route LLMs from preference data. In *The Thirteenth International Conference on Learning Representations*.

OpenAI. 2024. Learning to reason with llms.

Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, and Pengfei Liu. 2024. O1 replication journey: A strategic progress report – part 1. *Preprint*, arXiv:2410.18982.

Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, Peng Li, Wei Wei, Jing Shao, Chaochao Lu, Yue Zhang, Xian-Sheng Hua, Bowen Zhou, and Yu Cheng. 2025. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *Preprint*, arXiv:2503.21614.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, and Shiguo Lian. 2025. Dast: Difficulty-adaptive slow-thinking for large reasoning models. *Preprint*, arXiv:2503.04472.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *Preprint*, arXiv:2503.16419.

Liangtai Sun, Yang Han, Zihan Zhao, Da Ma, Zhennan Shen, Baocai Chen, Lu Chen, and Kai Yu. 2024.

Scieval: A multi-level large language model evaluation benchmark for scientific research. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19053–19061.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, and 75 others. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *Preprint*, arXiv:2501.12599.

NovaSky Team. 2025a. Sky-t1: Train your own o1 preview model within $450. https://novasky-ai.github.io/posts/sky-t1. Accessed: 2025-01-09.

NovaSky Team. 2025b. Think less, achieve more: Cut reasoning costs by 50 https://novasky-ai.github.io/posts/reduce-overthinking. Accessed: 2025-01-23.

Xiaoyu Tian, Sitong Zhao, Haotian Wang, Shuaiting Chen, Yunjie Ji, Yiping Peng, Han Zhao, and Xiangang Li. 2025. Think twice: Enhancing llm reasoning by scaling multi-round test-time thinking. *Preprint*, arXiv:2503.19855.

Armin Toroghi, Willis Guo, Ali Pesaranghader, and Scott Sanner. 2024. Verifiable, debuggable, and repairable commonsense logical reasoning via LLM-based theory resolution. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6634–6652, Miami, Florida, USA. Association for Computational Linguistics.

Yuxuan Wan, Wenxuan Wang, Yiliu Yang, Youliang Yuan, Jen-tse Huang, Pinjia He, Wenxiang Jiao, and Michael Lyu. 2024. LogicAsker: Evaluating and improving the logical reasoning ability of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2124–2155, Miami, Florida, USA. Association for Computational Linguistics.

Chenlong Wang, Yuanning Feng, Dongping Chen, Zhaoyang Chu, Ranjay Krishna, and Tianyi Zhou. 2025a. Wait, we don't need to "wait"! removing thinking tokens improves reasoning efficiency. *Preprint*, arXiv:2506.08343.

Jikai Wang, Juntao Li, Lijun Wu, and Min Zhang. 2025b. Efficient reasoning for llms through speculative chain-of-thought. *Preprint*, arXiv:2504.19095.

Rui Wang, Hongru Wang, Boyang Xue, Jianhui Pang, Shudong Liu, Yi Chen, Jiahao Qiu, Derek Fai Wong, Heng Ji, and Kam-Fai Wong. 2025c. Harnessing the reasoning economy: A survey of efficient reasoning for large language models. *Preprint*, arXiv:2503.24377.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. 2025. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. *Preprint*, arXiv:2503.10460.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024. Scaling inference computation: Compute-optimal inference for problem-solving with language models. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

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

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Zheng Lin, Li Cao, and Weiping Wang. Dynamic early exit in reasoning models. *Preprint*, arXiv:2504.15895.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *Preprint*, arXiv:2502.03387.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. Demystifying long chain-of-thought reasoning in llms. *Preprint*, arXiv:2502.03373.

Shuo Yin, Weihao You, Zhilong Ji, Guoqiang Zhong, and Jinfeng Bai. 2024. MuMath-code: Combining tool-use large language models with multi-perspective data augmentation for mathematical reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4770–4785, Miami, Florida, USA. Association for Computational Linguistics.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *Preprint*, arXiv:2503.18892.

Chong Zhang, Yue Deng, Xiang Lin, Bin Wang, Dianwen Ng, Hai Ye, Xingxuan Li, Yao Xiao, Zhanfeng Mo, Qi Zhang, and Lidong Bing. 2025a. 100 days after deepseek-r1: A survey on replication studies and more directions for reasoning language models. *Preprint*, arXiv:2505.00551.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, Irwin King, Xue Liu, and Chen Ma. 2025b. A survey on test-time scaling in large language models: What, how, where, and how well? *Preprint*, arXiv:2503.24235.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

A In-domain Evaluation

Table 5 presents the in-domain evaluation results on the **AIME25** and **AMC23** dataset, where the Qwen2.5-32B-Instruct model was trained on the s1K-1.1 and OpenThoughts-2K datasets using both the standard SFT and LS-Mixture SFT methods, respectively.

Table 5: Evaluation on AIME25 and AMC23

Method	**AIME25**		**AMC23**	
	Acc↑	Length↓	Acc↑	Length↓
s1K-1.1				
SFT	43.3	55,038.3	87.5	22,547.6
LS-Mixture SFT	50.0	44,540.2	95.0	14,329.0
OpenThoughts-2K				
SFT	33.3	58694.5	85.0	22,684.5
LS-Mixture SFT	46.7	32842.6	92.5	15,000.5

B Out-of-distribution Evaluation

Table 6 presents the out-of-distribution evaluation result on the **MBPP** and **HumanEval** dataset, where the Qwen2.5-32B-Instruct model was trained on the s1K-1.1 and OpenThoughts-2K datasets using both the standard SFT and LS-Mixture SFT methods, respectively.

Table 6: Evaluation on MBPP and HumanEval

Method	**MBPP**		**HumanEval**	
	Acc↑	Length↓	Acc↑	Length↓
s1K-1.1				
SFT	72.0	21.909.7	66.5	20130.3
LS-Mixture SFT	75.2	15,007.8	69.5	16,417.1
OpenThoughts-2K				
SFT	73.2	21,253.5	68.3	18133.7
LS-Mixture SFT	77.2	13,756.0	70.7	11841.3

C The Generalizability of Model Size

The main text evaluates the performance of the 32B model variant. To demonstrate the generalizability of our method across different model scales, this section additionally presents the results of 7B and 14B models under both standard SFT and LS-Mixture SFT methods in s1K-1.1 dataset. Tables 7, 8, and 9 present the evaluation results on three benchmarks: MATH500, AIME24, and GPQA, respectively. These experimental findings consistently align with our observations and conclusions.

Table 7: The performance of Qwen2.5-Instruct 7B and 14B models on **MATH500** after fine-tuning on the s1K-1.1 dataset using both standard SFT and LS-Mixture SFT.

	Method	**MATH500**	
		Acc↑	Length↓
7B	SFT	77.8	22,705.0
	LS-Mixture SFT	81.6	17,890.5
14B	SFT	88.2	17,917.0
	LS-Mixture SFT	91.2	12,559.3

Table 8: The performance of Qwen2.5-Instruct 7B and 14B models on **AIME24** after fine-tuning on the s1K-1.1 dataset using both standard SFT and LS-Mixture SFT.

	Method	**AIME24**	
		Acc↑	Length↓
7B	SFT	16.7	65,304.8
	LS-Mixture SFT	20.0	43,470.0
14B	SFT	30.0	64,532.4
	LS-Mixture SFT	40.0	37,958.1

Table 9: The performance of Qwen2.5-Instruct 7B and 14B models on **GPQA** after fine-tuning on the s1K-1.1 dataset using both standard SFT and LS-Mixture SFT.

	Method	**GPQA**	
		Acc↑	Length↓
7B	SFT	37.9	79,867.3
	LS-Mixture SFT	44.4	54,802.5
14B	SFT	51.0	58,442.7
	LS-Mixture SFT	54.5	30,376.6

D Evaluation Detail

We evaluate the performance of models on these benchmarks using the LightEval (Fourrier et al., 2023) framework following the open-r1 project (HuggingFace, 2025). In order to eliminate randomness in the evaluation, the presented accuracy corresponds to the median of five independent repeated experiments.

E Prompt Template

E.1 The prompt of Rewriter Model

Rewriter Model

You have a QUESTION and a THOUGHT PROCESS now, and you need to simplify the THOUGHT PROCESS while maintaining its original structure and steps.

QUESTION: {question}

THOUGHT PROCESS: {thought_process}

Now, you need to simplify the THOUGHT PROCESS while maintaining its original structure and steps. For each step in the original THOUGHT PROCESS:
1. Keep the original logical flow and steps as much as possible, including the thinking process, verification process, and the final answer.
2. Remove redundant tokens.
3. Preserve the step-by-step format.
4. Allow condensed thought processes to include attempts at different reasoning processes.
Do not add any new information that wasn't in the original THOUGHT PROCESS.

SIMPLIFIED THOUGHT PROCESS:

E.2 The prompt of detailed thinking mode

Detail Thinking Mode

Answer the problem with a **detailed** thinking process:

E.3 The prompt of brief thinking mode

Brief Thinking Mode

Answer the problem with a **brief** thinking process:

E.4 The prompt of balanced thinking mode

Balanced Thinking Mode

Answer the problem with a **appropriate** (between detailed and brief) thinking process:

E.5 The prompt of Direct Compression

Direct Compression

You have a question now:

QUESTION:
{question}

THOUGHT PROCESS:
{thought_process}

Now, you need to simplify the THOUGHT PROCESS as short as possible to only include the key information needed to solve the question. And do not add additional information that is not included in the original THOUGHT PROCESS.

SIMPLIFIED THOUGHT PROCESS:

F Dataset Profile

Table 10: The statistical profile of the datasets used in this study, namely s1K-1.1 and s1K-mix. For each dataset, we report the number of rows and the average text length.

Dataset	Num of Rows	Average Length
s1K-1.1	1000	29667.49
s1K-mix	1984	17406.11

G Training Hyperparameters

All experiments were run in a GPU cluster of 16 * A800. The hyperparameters used for training are presented in Table 11, while any parameters not explicitly specified utilize the default values provided by LlamaFactory (Zheng et al., 2024).

Table 11: Training Hyperparameters

Hyperparameter	Value
cutoff_len	4096
learning_rate	1e-5
lr_scheduler_type	cosine
warmup_ratio	0.05
bf16	true
optimizer	AdamW
weight_decay	1e-4

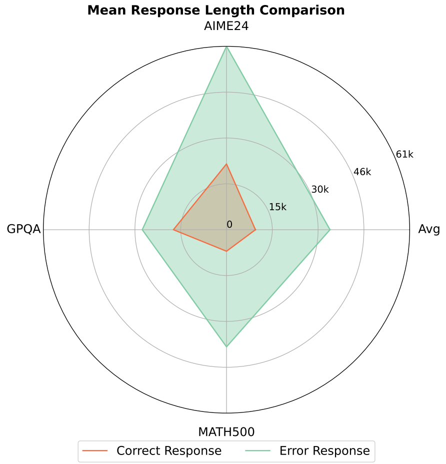


Figure 4: Comparison of mean response lengths for correct (red) and error (green) predictions of the **s1-mix-32B** model across the MATH500, GPQA, and AIME24 benchmarks.

H Analysis of Response Length

Our analysis of the evaluation results from the **s1-mix-32B** reveals a correlation between response correctness and length. As illustrated in Figure 4, incorrect responses demonstrate greater verbosity across all evaluation datasets. Specifically, the average response length of incorrect examples is approximately 3.58 times longer than the correct part. This finding suggests that when the model is uncertain or unable to produce an accurate answer, it tends to generate more unhelpful text.

I Case Study of Rewriting

Figure 5 illustrates an example of applying a structure-preserving rewriting strategy to a long CoT in the s1K-1.1 dataset, resulting in a concise short CoT. As evidenced, the rewritten version maintains identical logical structure and correctness while significantly improving expression conciseness.

J Word cloud of Datasets

Figures 6 and 7 respectively display word clouds of the chain-of-thought trajectories from our experiments on the long reasoning dataset and the short reasoning dataset. As can be observed from these two figures, the distribution of common words undergoes a significant change following structure-preserved rewriting, notably marked by the disappearance of the words 'wait' and 'need'.

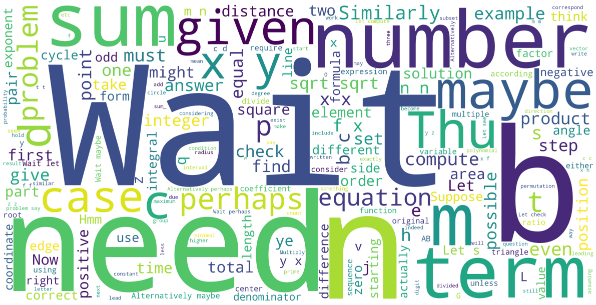


Figure 6: Word cloud of CoT trajectories in the long reasoning dataset.

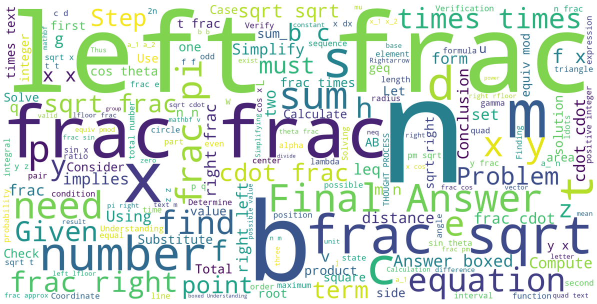
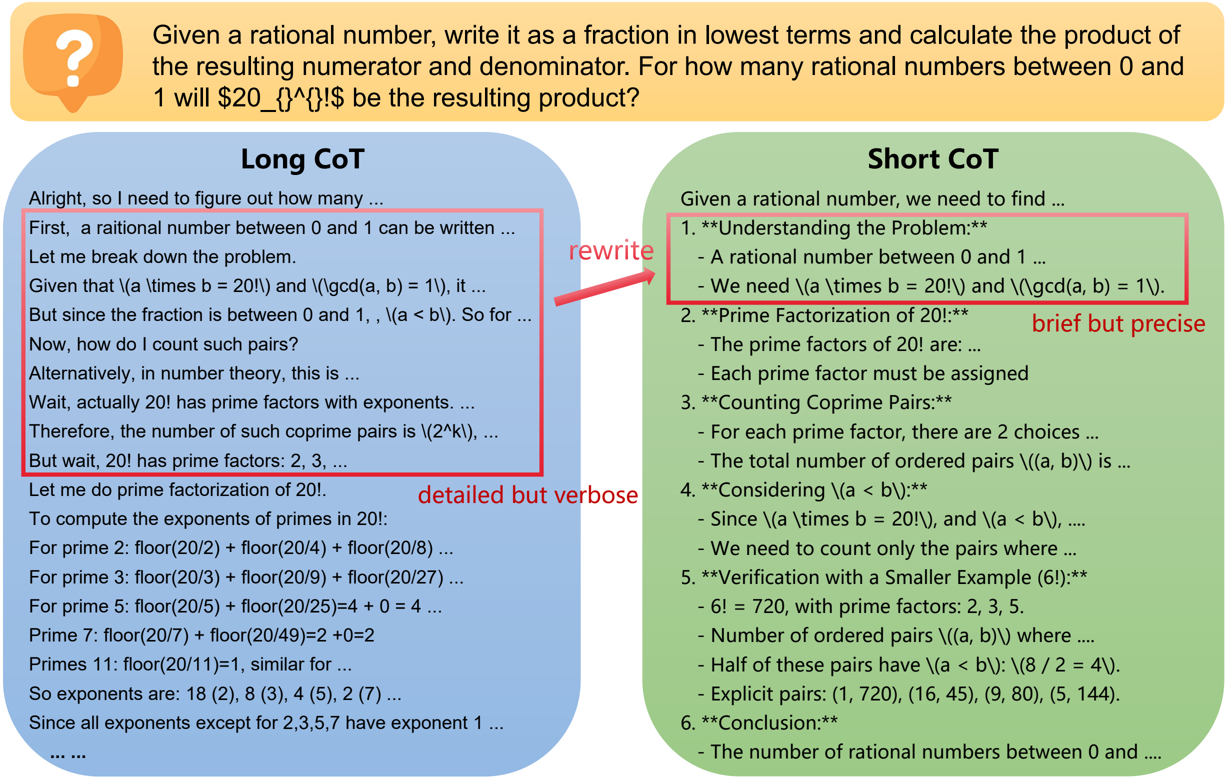


Figure 7: Word cloud of CoT trajectories in the short reasoning dataset.



Given a rational number, write it as a fraction in lowest terms and calculate the product of the resulting numerator and denominator. For how many rational numbers between 0 and 1 will $20_{}^{}!$ be the resulting product?

Long CoT

Alright, so I need to figure out how many ...
First, a raitional number between 0 and 1 can be written ...
Let me break down the problem.
Given that \(a \times b = 20!\) and \(\gcd(a, b) = 1\), it ...
But since the fraction is between 0 and 1, , \(a < b\). So for ...
Now, how do I count such pairs?
Alternatively, in number theory, this is ...
Wait, actually 20! has prime factors with exponents. ...
Therefore, the number of such coprime pairs is \(2^k\), ...
But wait, 20! has prime factors: 2, 3, ...
Let me do prime factorization of 20!.
To compute the exponents of primes in 20!:
For prime 2: floor(20/2) + floor(20/4) + floor(20/8) ...
For prime 3: floor(20/3) + floor(20/9) + floor(20/27) ...
For prime 5: floor(20/5) + floor(20/25)=4 + 0 = 4 ...
Prime 7: floor(20/7) + floor(20/49)=2 +0=2
Primes 11: floor(20/11)=1, similar for ...
So exponents are: 18 (2), 8 (3), 4 (5), 2 (7) ...
Since all exponents except for 2,3,5,7 have exponent 1 ...
... ...

Short CoT

Given a rational number, we need to find ...
1. **Understanding the Problem:**
 - A rational number between 0 and 1 ...
 - We need \(a \times b = 20!\) and \(\gcd(a, b) = 1\).
2. **Prime Factorization of 20!:**
 - The prime factors of 20! are: ...
 - Each prime factor must be assigned
3. **Counting Coprime Pairs:**
 - For each prime factor, there are 2 choices ...
 - The total number of ordered pairs \((a, b)\) is ...
4. **Considering \(a < b\):**
 - Since \(a \times b = 20!\), and \(a < b\), ...
 - We need to count only the pairs where ...
5. **Verification with a Smaller Example (6!):**
 - 6! = 720, with prime factors: 2, 3, 5.
 - Number of ordered pairs \((a, b)\) where ....
 - Half of these pairs have \(a < b\): \(8 / 2 = 4\).
 - Explicit pairs: (1, 720), (16, 45), (9, 80), (5, 144).
6. **Conclusion:**
 - The number of rational numbers between 0 and ....

Figure 5: An example of applying a structure-preserving rewriting strategy to transform a long CoT from the s1K-1.1 dataset into a concise short CoT. As evidenced, the rewritten version maintains identical logical structure and correctness while significantly improving expression conciseness.