Not All Tokens Are What You Need In Thinking

Anonymous ACL submission

Abstract

Modern reasoning models, such as OpenAI's o1 and DeepSeek-R1, exhibit impressive problem-solving capabilities but suffer from critical inefficiencies: high inference latency, excessive computational resource consumption, and a tendency toward overthinking—generating verbose chains of thought (CoT) laden with redundant tokens that contribute minimally to the final answer. To address these issues, we propose Conditional Token Selection (CTS), a token-level compression framework with a flexible and variable compression ratio that identifies and preserves only the most essential tokens in CoT. CTS evaluates each token's contribution to deriving correct answers using conditional importance scoring, then trains models on compressed CoT. Extensive experiments demonstrate that CTS effectively compresses long CoT while maintaining strong reasoning performance. Notably, on the GPQA benchmark, Qwen2.5-14B-Instruct trained with CTS achieves a 9.1% accuracy improvement with 13.2% fewer reasoning tokens (13% training token reduction). Further reducing training tokens by 42% incurs only a marginal 5% accuracy drop while yielding a 75.8% reduction in reasoning tokens, highlighting the prevalence of redundancy in existing CoT.

1 Introduction

Large reasoning models such as o1(Jaech et al., 2024) and R1(Guo et al., 2025) significantly enhance their reasoning capabilities through reinforcement learning, instructing models to generate thoughtful reasoning steps before producing final answers. Guo et al. (2025) demonstrated that by fine-tuning non-reasoning models like Qwen2.5-14B-Instruct on long Chain of Thought (CoT) data generated by R1, these models can acquire comparable reasoning abilities, even surpassing o1-mini on math and code reasoning tasks. Consequently, numerous distilled R1 reasoning datasets have emerged, including s1K, SkyThought, OpenMathReasoning, and AM-1.4M (Team, 2025; Zhao et al., 2025; Muennighoff et al., 2025; Moshkov et al., 2025). Small language models trained on these datasets consistently demonstrate remarkable reasoning capabilities. However, the ever-increasing length of CoT sequences burdens both training and inference, with recent studies (Sui et al., 2025) revealing that models often overthink, expending substantial resources on redundant reasoning steps.

This inefficiency raises a critical question: How can we preserve the accuracy gains of long CoT reasoning while eliminating its computational waste? Existing solutions, such as TokenSkip's task-agnostic compression (Xia et al., 2025), show promise for short CoT sequences but fail to address the unique challenges of reinforcement learning-generated long CoT data (spanning thousands of tokens). Moreover, they overlook contextual signals like questions and answers, which prior work Tian et al. (2025) identifies as key to effective compression.

To bridge this gap, we propose Conditional Token Selection (CTS), a framework that dynamically prunes redundant reasoning tokens while preserving those essential for deriving answers. CTS leverages a Reference Model (RM) trained on high-quality reasoning corpora to score token importance conditioned on critical context (e.g., questions and answers). As shown in Figure 1, by filtering CoT data at adjustable compression ratios and then fine-tuning model with compressed data, we enable models to learn how to skip unnecessary reasoning tokens during inference.

We conducted extensive experiments on models of various sizes, including the LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024) and the Qwen2.5-Instruct series (Qwen et al., 2025). The experimental results demonstrate the effectiveness of our method and confirm that there indeed exist

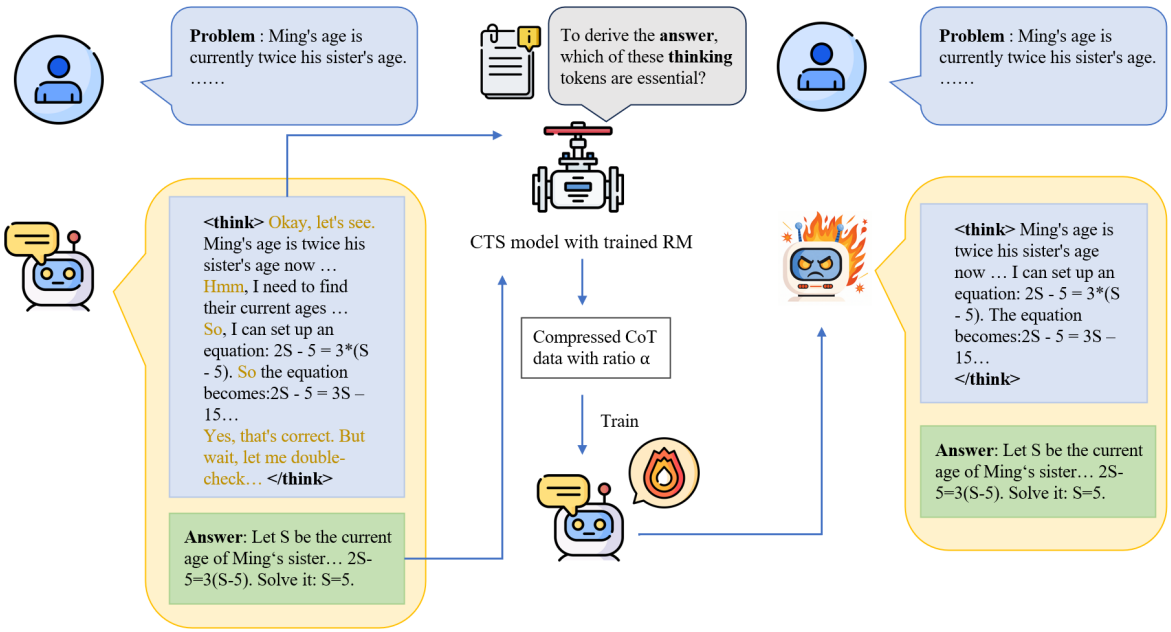

Figure 1: Illustration of Conditional Token Selection (CTS). For long CoT datasets, CTS leverages a well-trained Reference Model (RM) to evaluate the importance of each thinking token conditional on the answer, removing less important tokens based on the compression ratio α. The model is then trained on this compressed data, enabling more efficient reasoning capabilities.

many redundant thinking tokens in long CoT data. Notably, for Qwen2.5-14B-Instruct on the GPQA benchmark, CTS achieves a **9.1%** accuracy gain with 13.2% fewer reasoning tokens when reducing training tokens by 13%. Further reducing training tokens by **42%** leads to a marginal 5% accuracy drop but yields a substantial **75.8%** reduction in reasoning tokens. On other benchmarks, such as MATH500 and AIME24, as well as with other models, using the compressed training data obtained through CTS resulted in improved accuracy compared to the original data after training.

In summary, our key contributions are:

- We introduce the Conditional Token Selection framework, which assigns conditional importance scores to tokens within CoT trajectories based on critical contextual information, thereby selectively preserving essential reasoning tokens necessary for accurate answer derivation at adjustable compression ratios.
- We provide a Reference Model (RM) trained on high-quality reasoning corpora that is capable of judging token importance in reasoning CoTs, along with methods for corpus filtering. This model can be applied to other independent tasks, such as prompt compression.
- We comprehensively compare token-based compression methods, including both conditional and non-conditional approaches, for long CoT data obtained through reinforcement learning, validating the effectiveness of token selection strategies.

2 Preliminaries

In this section, we introduce some important preliminary concepts.

2.1 Token Compression based on Perplexity

For a given context $\boldsymbol{x}_{t-1} = \{x_i\}_{i=1}^{t-1}$, the self-information of a token x_t can be defined as:

$$I(x_t) = -\log_2 P(x_t \mid \boldsymbol{x}_{t-1}) \quad (1)$$

Perplexity (PPL) is then defined based on self-information as:

$$\mathrm{PPL}(x_t) = 2^{I(x_t)} \quad (2)$$

Perplexity is commonly used to measure a language model's ability to predict the given context. Removing tokens with lower perplexity (Li et al., 2023) has a relatively small impact on the model's understanding and prediction of the context.

2.2 Conditional and Unconditional Compression

To address information redundancy in long contexts and the issue of resource consumption during

inference, Li et al. (2023) proposed a method that uses a small language model to calculate the importance of each lexical unit (such as sentences or tokens) in the original prompts, and then drops the less informative content for prompt compression. Subsequent studies by Jiang et al. (2023, 2024) followed this line of research, proposing more fine-grained compression methods. Pan et al. (2024) utilized a BERT model to transform prompt compression into a classification problem.

These compression methods can be categorized into unconditional and conditional approaches, based on whether important information is used as a condition during compression. For example, user queries and instructions can serve as important conditions for compressing instances within a prompt. Specifically, during compression, unconditional methods directly calculate metrics like self-information or perplexity as in Equation (1) and (2). In contrast, conditional methods determine token importance based on crucial conditional information v:

$$I(x_t \mid v, \boldsymbol{x}_{t-1}) = -\log_2 P(x_t \mid v, \boldsymbol{x}_{t-1}) \quad (3)$$

In terms of effectiveness, conditional methods generally outperform unconditional methods because they preserve more information (Pan et al., 2024).

3 Conditional Token Selection

Conditional Token Selection (CTS) builds upon the idea of conditional token compression in prompt. It aims to compress long CoT sequences, such as those generated by harnessing reinforcement learning techniques (e.g., R1 (Sui et al., 2025)), and subsequently fine-tune models on this compressed data. The goal is to enhance model performance while reducing training and inference resource consumption.

3.1 Problem Formulation

Given a long CoT dataset, where each instance $\boldsymbol{x}$ consists of a problem $\boldsymbol{x}^{\text{prob}}$, thinking tokens $\boldsymbol{x}^{\text{thk}}$, and a final answer $\boldsymbol{x}^{\text{ans}}$, denoted as $\boldsymbol{x} = \{\boldsymbol{x}^{\text{prob}}, \boldsymbol{x}^{\text{thk}}, \boldsymbol{x}^{\text{ans}}\}$ as depicted in Figure 1. Let us consider a small language model, such as Qwen2.5-7B-Instruct, with its original parameters denoted as θ_{LM}.

The distillation objective is to train this small model to imbue it with reasoning (thinking) capabilities and achieve strong performance by minimizing the following loss function:

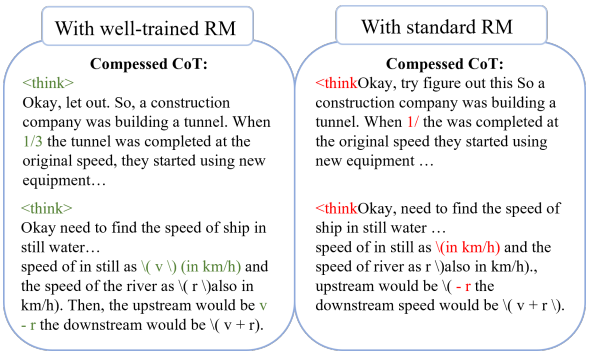

Figure 2: An example of CoT compression using CTS, where the left figure employs a well-trained RM, while the right uses a standard RM.

$$\mathcal{L} = -\sum_{i=1}^{l} \log P_{\theta_{\text{LM}}}(y_i \mid \boldsymbol{x}^{\text{prob}}) \quad (4)$$

where $\boldsymbol{y} = \{y_i\}_{i=1}^{l} = \boldsymbol{x}^{\text{thk}} \oplus \boldsymbol{x}^{\text{ans}}$ is the target sequence of l tokens, with $\oplus$ denoting the concatenation of tokens.

The objective of a long CoT compression system can be defined as:

$$\min_{\tilde{\boldsymbol{y}}} \text{dist}(A, \tilde{A}) + \lambda \|\tilde{\boldsymbol{y}}\|_0, \quad (5)$$

Where $\tilde{\boldsymbol{y}}$ represents the compressed CoT, a subsequence of $\boldsymbol{y}$. A and $\tilde{A}$ represent, respectively, the answers to any question Q given by the small language models trained with $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$. Here, $\text{dist}(\cdot, \cdot)$ is a function measuring the distance (e.g., KL divergence). λ serves as a hyper-parameter balancing the compression ratio. $\|\cdot\|_0$ is a penalty.

3.2 Reference Modeling

When determining token importance in CoT data, we typically need a Reference Model, which is usually a lightweight small language model. However, through experimental observations, we found that using small language models directly to compress CoT tends to remove important but commonly used numbers or alphanumeric symbols. As shown in the right panel of Figure 2, the water flow velocity variable v has been removed. This variable is crucial for understanding the subsequent equations.

To teach the Reference Model which numbers and reasoning symbols are important for reaching the final answer, we curated a high-quality dataset $\{\boldsymbol{q}, z_1, \ldots, z_K, \boldsymbol{a}\}$ that reflects the desired data distribution. We then train a reference model (RM)

using cross-entropy loss on the curated data.

$$\mathcal{L}_{RM} = -\sum_{i=1}^{K} \log P(z_i \mid x^{ins}, \boldsymbol{q}, \boldsymbol{a}) \quad (6)$$

Where x^{ins} represents "*For a problem* $\boldsymbol{q}$*, the following reasoning steps are important to get the answer* $\boldsymbol{a}$". The resulting RM is then used to assess the token importance within long CoT trajectories, allowing us to focus on the most influential tokens in the training process.

3.3 Token Selection Based on Conditional Perplexity Differences

To perform conditional compression on Chain-of-Thought data, we need to evaluate the conditional importance of each token in the target sequences y.

To mitigate the inaccuracy introduced by the conditional independence assumption, we adopt the iterative compression method from Jiang et al. (2023), where we first divide y into several segments $\mathcal{S} = \{s_1, \ldots, s_m\}$, then compress within each segment to obtain $\{\tilde{s}_1, \ldots, \tilde{s}_m\}$, and finally concatenate the compressed segments to form the final compressed text.

For more fine-grained assessment of token conditional importance, we use the distribution shift caused by the condition of the answer to represent the association between the thinking token and the answer. Thus, we can derive a score r_i for each token in the target sequence y, calculated as follows:

$$r_i = \text{PPL}(x_i^{\text{thk}} \mid x_{<i}^{\text{thk}}) - \text{PPL}(x_i^{\text{thk}} \mid \boldsymbol{x}^{ans}, x_{<i}^{\text{thk}})$$

Finally, given a compression ratio α, after scoring each token in the reasoning chain using the RM, we determine a threshold r_α represented by the α-quantile of importance scores, thereby selecting thinking tokens whose scores exceed the threshold:

$$\tilde{\boldsymbol{x}}^{thk} = \{x_i^{\text{thk}} \mid r_i > r_\alpha\}$$

4 Experiments

4.1 Experimental Setup

Reference Model Training To train our mathematical reasoning reference model, we utilized the first 9.3K MATH training set from Face (2025), where problems originate from NuminaMath 1.5 and reasoning traces were generated by DeepSeek. To teach the model to evaluate token importance in mathematical reasoning chains, we employed carefully designed prompts (see Appendix) to select 8M tokens from an original 54M reasoning tokens.

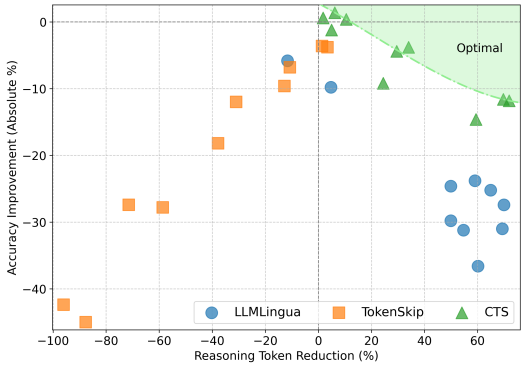

Figure 3: Performance metrics (Reasoning Token Reduction percentage and Absolute Accuracy Improvement (%) relative to original) for various compression methods and ratios on Qwen2.5-7B-Instruct and Qwen2.5-14B-Instruct models and MATH500. The top-right region represents optimal performance, signifying higher accuracy and reduced reasoning token usage.

Implementation Details & Dataset To demonstrate the effectiveness of our Reference Model (RM) and Conditional Token Selection framework, we leveraged the framework proposed by (Jiang et al., 2024), employing our trained RM for Conditional Token Selection on the second 9.3K MATH training set from Face (2025). We then fine-tuned LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024) and Qwen2.5-7B-Instruct and Qwen2.5-14B-Instruct (Qwen et al., 2025) using the compressed dataset with compression ratios α of $\{0.5, 0.6, 0.7, 0.8, 0.9\}$.

Table 1: Detailed Information of the Datasets

Dataset	Size
OpenMath	18,600
MATH500	500
AIME2024	30
GPQA Diamond	198

Evaluation Benchmarks & Metrics The evaluation leverages three widely used reasoning benchmarks: **AIME24**, **MATH500**, and **GPQA Diamond** (Mathematical Association of America, 2024; Hendrycks et al., 2021; Rein et al., 2024). We used the actual compression ratio, average accuracy, and average reasoning token count as metrics

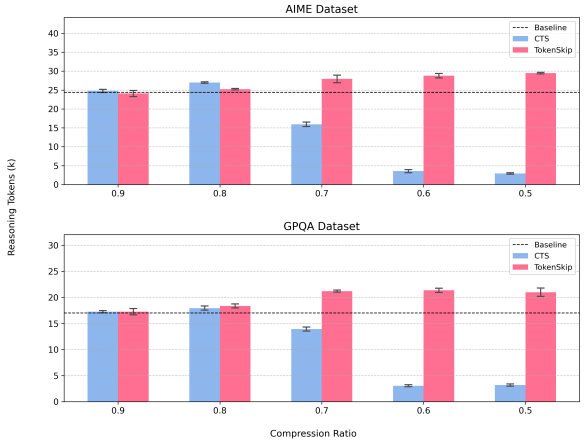

Figure 4: Compression Ratio vs. Average Reasoning Tokens for CTS and TokenSkip on the Qwen2.5-7B-Instruct model across different test sets.

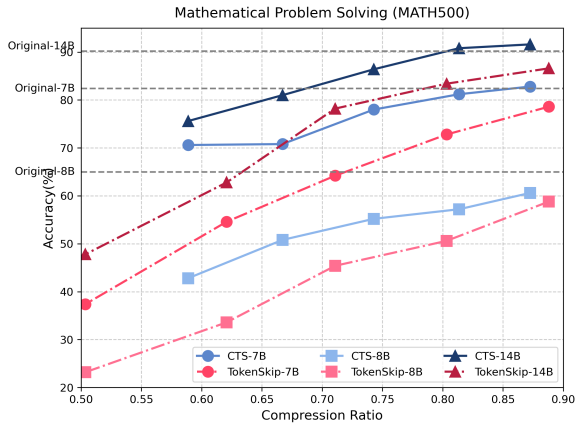

Figure 5: Comparison of accuracy across different compression ratios for various model configurations.

to evaluate compression methods. All training and evaluation were conducted on 8 NVIDIA A800 GPUs. During training, we fine-tuned for 3 epochs with a batch size of 16. The maximum learning rate was set at 1e-5 with a cosine decay schedule. We set the maximum sequence length to 4096.

Baselines In our main experiments, we compared Conditional Token Selection with **unconditional TokenSkip** (Xia et al., 2025), **LLMLingua** (Jiang et al., 2023) and **Prompt-based Compression**. We designate the method that directly uses the original CoT data for training as **Original**. For the Prompt-based method, we instructed GPT-4o[1] to compress long CoT reasoning by providing prompts such as "Please retain important reasoning tokens in the Chain-of-Thought and remove unnecessary ones, preserving α% of the original tokens." However, we observed that GPT tends to be overly aggressive in compression, consistently preserving less than 10% of the original tokens regardless of the specified α value. Therefore, we did not set a specific compression ratio α and simply used GPT-4o to compress the reasoning chains directly. These baselines are referred to as **GPT-4o**, **LLMlingua** and **TokenSkip** in Table 2, respectively.

4.2 Main Results

Table 2 and 4 presents the performance of different compression methods on the Qwen2.5-14B-Instruct model across various compression ratios. Notably, our method achieves the highest accuracy across all five compression ratios compared to other compression approaches. Figure 5 illustrates the accuracy of CTS versus TokenSkip across different models and compression ratios. CTS consistently outperforms TokenSkip, highlighting the superiority of conditional token compression.

Although the Prompt-based GPT-4o shown in Table 2 uses fewer inference tokens, this is a result of its excessive compression of the chain of thought. In reality, the model fails to learn thinking capabilities from long CoT data, resulting in low accuracy across all three benchmarks.

Furthermore, for CTS, when the compression ratio is 0.9 and 0.8, the model shows improvements in both accuracy and inference efficiency. For example, on AIME24, at compression ratios of 0.9 and 0.8, accuracy increased by 10% and 4.3% respectively, while inference tokens decreased by 1373 and 1274. Notably, for GPQA Diamond, CTS achieves a 9.1% accuracy gain with 13.2% fewer reasoning tokens. Further reduction of inference tokens by 75.8% results in only a 5% accuracy drop, and since the compression ratio at this point is 42%, training costs are significantly reduced.

As the compression ratio decreases, although CTS accuracy declines compared to Original, inference tokens continue to decrease. Thus, there exists a trade-off between accuracy and inference efficiency. From Table 2, we can infer that the optimal point lies between ratios 0.7 and 0.8, where model capability remains unchanged while minimizing inference token consumption.

However, poor compression methods can actually decrease model inference efficiency. As shown in Figure 4, as the compression ratio increases, TokenSkip's token consumption actually increases on the 7B model. This demonstrates that tokens in

[1]We use the gpt-4o-2024-08-06 version for experiments.

Table 2: Experimental results of various compression methods on **Qwen2.5-14B-Instruct**, showing accuracy, average reasoning CoT tokens, and compression ratio (actual ratio).

Methods	Ratio (Actual)	MATH500		AIME24		GPQA Diamond	
		Accuracy ↑	Tokens ↓	Accuracy ↑	Tokens ↓	Accuracy ↑	Tokens ↓
Original	1.0	90.2	5012	40	23041	51.5	12000
GPT-4o	0.9(0.06)	61.4	**283**	0	**353**	35.8	**353**
LLMLingua	0.9(0.88)	84.4	5597	33.3	19731	53.0	10689
	0.8(0.80)	65.6	2510	10.0	4230	44.9	4037
	0.7(0.71)	60.4	2511	6.7	4588	43.9	3371
	0.6(0.62)	59.0	2270	6.7	3076	40.9	3347
	0.5(0.50)	53.6	1998	3.3	3789	40.4	2796
TokenSkip	0.9(0.88)	86.6	4941	40.0	19985	50.0	12455
	0.8(0.80)	83.4	5549	26.7	20945	50.0	13275
	0.7(0.71)	78.2	6566	16.7	24718	43.4	15531
	0.6(0.62)	62.8	8595	10.0	27748	38.8	16764
	0.5(0.50)	47.8	9824	3.3	26555	31.8	19121
CTS	0.9(0.87)	**91.6**	4703	**50.0**	21668	**60.6**	10413
	0.8(0.81)	**90.8**	4922	**43.4**	21767	**53.5**	13136
	0.7(0.74)	**86.4**	3310	**33.3**	10448	**57.1**	10372
	0.6(0.66)	**81.0**	3787	**16.7**	10308	**48**	9712
	0.5(0.58)	**75.6**	2036	**10.0**	3196	**46.5**	2906

Table 3: Ablation Study Variant Comparison

Model Variant	**Conditional**	**RM**
Base	×	×
+ Conditional	✓	×
+ RM-Tuned	×	✓
Proposed (CTS)	✓	✓

CoT cannot be removed arbitrarily, which aligns with intuition.

Figure 3 displays the percentage of reasoning token reduction and accuracy improvement for three methods. Higher accuracy with greater reasoning token reduction is preferable. Therefore, our method achieves the optimal balance between token reduction and accuracy improvement.

4.3 Ablation Study

Our ablation study aims to verify: 1) The effectiveness of using a trained Reference Model (RM) for selecting valuable tokens, and 2) The effectiveness of conditional token importance assessment compared to unconditional methods. For the unconditional token importance assessment, we follow the framework established in Jiang et al. (2023).

We introduce the following variants of our method for the ablation study, as shown in Table 3: (1) Base: Using an untrained RM to predict token importance in CoT reasoning without conditioning; (2) + Conditional: Using an untrained RM to predict token importance in CoT reasoning with conditioning; (3) + RM-Tuned: Using a trained RM to predict token importance in CoT reasoning without conditioning; (4) Proposed (CTS): Using a well-trained RM to predict token importance in CoT reasoning with conditioning.

As shown in Figures 7 and 6, conditional compression methods play a crucial role in enhancing model capabilities. For the Qwen2.5-14B-Instruct model, the two curves with the highest accuracy in the figures utilized conditional token importance prediction methods. The Proposed method performs slightly better than the + Conditional approach, while the + RM-Tuned method shows marginal improvements over the Base method. This indicates that training the reference model provides some benefit in identifying important reasoning tokens. The modest improvement might be attributed to the limited size of the high-quality corpus used for training, which contained only 8M tokens.

Table 4: Experimental results of various compression methods on **Qwen2.5-7B-Instruct**, showing accuracy, average reasoning CoT tokens, and compression ratio (actual ratio).

Methods	Ratio (Actual)	MATH500		AIME24		GPQA Diamond	
		Accuracy ↑	Tokens ↓	Accuracy ↑	Tokens ↓	Accuracy ↑	Tokens ↓
Original	1.0	82.4	7244	20	24396	37.8	17038
LLMLingua	0.9(0.88)	72.6	6804	13.3	**23903**	39.8	**14470**
	0.8(0.80)	58.6	**2969**	6.7	**4194**	34.3	**4421**
	0.7(0.71)	57.2	**2542**	6.7	**3692**	32.8	**3236**
	0.6(0.62)	55.0	**2178**	3.3	**3084**	**33.3**	3203
	0.5(0.50)	51.4	2226	3.3	3603	30.8	**2462**
TokenSkip	0.9(0.88)	78.6	6997	**23.3**	24094	38.8	17263
	0.8(0.80)	72.8	8172	10.0	25223	39.3	18365
	0.7(0.71)	64.2	9984	6.6	27946	32.3	21219
	0.6(0.62)	54.6	11496	3.3	28802	26.2	21371
	0.5(0.50)	37.4	13595	0	29470	31.3	21012
CTS	0.9(0.87)	**82.8**	**6497**	20	24769	**43.4**	17272
	0.8(0.81)	**81.2**	6886	**23.3**	27006	**39.3**	17961
	0.7(0.74)	**78.0**	5109	**13.3**	15929	**42.4**	13937
	0.6(0.66)	**70.8**	2198	**10.0**	3550	32.3	**3055**
	0.5(0.58)	**70.6**	**2039**	**6.7**	**2993**	**32.8**	3187

Table 5 demonstrates that reasoning tokens do not differ substantially across variants, indicating that various methods do not significantly improve inference efficiency. In fact, at certain compression ratios, efficiency actually decreases. This suggests that accuracy improvements at the same compression ratio come with a corresponding increase in reasoning token consumption. This observation aligns with test time scaling results in Zhang et al. (2025), which indicate that model capability scales with inference length.

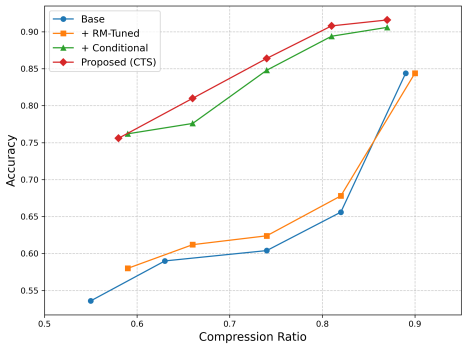

Figure 6: Ablation experiments of the Qwen2.5-14B-Instruct model on the MATH500 dataset under different compression ratios

5 Related Work

5.1 Overthinking in Long CoT Reasoning Models

Chen et al. (2025); Team et al. (2025) demonstrated that in long CoT reasoning models, models generate overly detailed or unnecessarily elaborate reasoning steps, ultimately reducing their problem-solving efficiency. Many current reasoning models with smaller parameter counts tend to produce verbose reasoning or redundant intermediate steps, making them unable to provide answers within the user-defined token budget. These results reveal the phenomenon of redundant thinking in reasoning models.

5.2 Efficient Reasoning

Prompt-based Chain of Thought (CoT) methods (Wei et al., 2022; Kojima et al., 2022) guide models to think step-by-step, enhancing their problem-solving capabilities. Chain of Draft (Xu et al., 2025), through prompting, retains essential formulas and numbers in the thought chain, maintaining performance while reducing inference costs. Lee et al. (2025) conducted a comprehensive comparison of prompt-based CoT compression methods. Wu et al. (2025); Ma et al. (2025) implement thought intervention by incorporating first-person prompts in the model's thinking process,

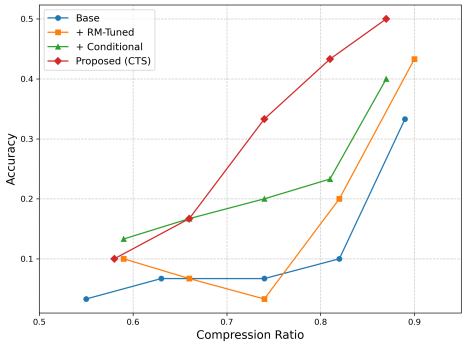

Figure 7: Ablation experiments of the Qwen2.5-14B-Instruct model on the AIME24 dataset under different compression ratios

Table 5: Comparison of Different Methods Across Varying Ratios. Cell values are shown as: Absolute Value (Corresponding Ratio).

Ratio	Base	+RM	+Conditional	Proposed
0.9	5023	5597	4563	5012
0.8	2510	2992	5001	4703
0.7	2511	2369	2883	3310
0.6	2270	2167	1952	3787
0.5	1998	2036	1787	2036

such as "<think> I think I have finished the thinking. </think>", to achieve instruction following and efficient reasoning.

Kang et al. (2025) improves reasoning efficiency by compressing the reasoning process. Yu et al. (2025) trains models by collecting variable-length CoT reasoning data with both long and short reasoning, thereby experimenting with reduced inference tokens. Munkhbat et al. (2025); Yeo et al. (2025); Xia et al. (2025) collect short CoT data by reducing redundant reasoning steps after full-length reasoning.

5.3 Prompt Compression

As language models increase in parameter scale, their capabilities also grow stronger, enabling many tasks to achieve better results merely by changing the model's input, such as RAG (Lewis et al., 2020) and few-shot learning (Wang et al., 2020). As prompts become longer, resource consumption increases significantly. Li et al. (2023) proposed evaluating token importance by calculating token perplexity in context to compress the input.

Building on this foundation, Jiang et al. (2023) proposed a mixed coarse-grained and fine-grained compression method. Jiang et al. (2024) extended this work by introducing a task-aware prompt compression method that incorporates conditional information from the query when calculating perplexity. Pan et al. (2024) transformed token compression into a binary classification problem, utilizing BERT models for compression.

6 Conclusion

We propose the Conditional Token Selection (CTS) method, which utilizes a fine-tuned Reference Model to calculate conditional perplexity differences for each token in long CoT data, identifying the most critical thinking tokens for deriving correct answers. By applying flexible compression ratios, our method compresses CoT reasoning data to enable more efficient training while maintaining the model's reasoning capabilities and ensuring output efficiency. Extensive experiments across various LLMs and tasks validate the effectiveness of CTS. Comprehensive ablation studies also demonstrate the importance of each component in our method. Impressively, our approach achieved up to a 10% improvement in accuracy while reducing reasoning tokens by 6% (Qwen2.5-14B-Instruct model on AIME24 with an actual compression ratio of 0.87). For inference efficiency, we achieved a maximum reduction of 75.8% in reasoning tokens with only a 5% accuracy drop (Qwen2.5-14B-Instruct model on GPQA Diamond with an actual compression ratio of 0.58).

Additionally, the RM trained on valuable reasoning tokens can function as a standalone model for other methods requiring assessment of reasoning token importance. This work contributes to making powerful reasoning capabilities more accessible in resource-constrained environments and opens new directions for developing efficient reasoning models.

Limitation

First, our approach is constrained by data limitations, as the quantity of valuable reasoning tokens used for training the Reference Model is insufficient for broader token importance assessment capabilities, especially in specialized domains like code-related problems. Resource constraints also prevented experiments with larger models such as 32B and 72B variants.

Second, our method focuses primarily on compressing existing reasoning patterns rather than developing new reasoning strategies, and requires

high-quality reasoning datasets which may not be available for all domains or tasks. The token importance evaluation, while effective, remains an approximation of each token's true contribution to the reasoning process.

Third, very high compression ratios may affect the interpretability of reasoning chains for human readers, potentially limiting their educational or explanatory value in applications where transparency is important. Additionally, while we demonstrate effectiveness across several reasoning benchmarks, these may not fully represent the complexity and diversity of real-world reasoning tasks.

Ethical Statement

The datasets used in our experiments are publicly available, English-labeled, and privacy-compliant, with all research artifacts licensed for permissible use. Our methodology adheres to ACL ethical guidelines. However, while our approach enhances reasoning efficiency, it may also accelerate AI deployment in sensitive areas without proper safeguards and reduce transparency in decision-making. We stress the need for responsible use, prioritizing transparency, fairness, and accountability—particularly in explainability-critical applications, where lower compression ratios may be necessary to preserve interpretability. In our humble opinion, we have not discerned any potential social risks.

References

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. 2025. Do not think that much for 2+3=? on the overthinking of o1-like llms. *Preprint*, arXiv:2412.21187.

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

Aaron Grattafiori, Abhimanyu Dubey, and Abhinav Jauhri et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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. *arXiv preprint arXiv:2103.03874*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.

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

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

Ayeong Lee, Ethan Che, and Tianyi Peng. 2025. How well do llms compress their own chain-of-thought? a token complexity approach. *arXiv preprint arXiv:2503.01141*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Yucheng Li, Bo Dong, Chenghua Lin, and Frank Guerin. 2023. Compressing context to enhance inference efficiency of large language models. *arXiv preprint arXiv:2310.06201*.

Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. 2025. Reasoning models can be effective without thinking. *Preprint*, arXiv:2504.09858.

Mathematical Association of America. 2024. Aime. URL `https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions/`.

Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. 2025. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset. *arXiv preprint arXiv:2504.16891*.

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.

Tergel Munkhbat, Namgyu Ho, Seo Hyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. 2025. Self-training elicits concise reasoning in large language models. *Preprint*, arXiv:2502.20122.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, and et al. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981, Bangkok, Thailand. Association for Computational Linguistics.

Qwen, :, and An Yang et al. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Hanjie Chen, Xia Hu, and 1 others. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, and et al. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *Preprint*, arXiv:2501.12599.

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

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. *arXiv preprint arXiv:2503.19855*.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H 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*.

Tong Wu, Chong Xiang, Jiachen T. Wang, and Prateek Mittal. 2025. Effectively controlling reasoning models through thinking intervention. *Preprint*, arXiv:2503.24370.

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

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

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

Bin Yu, Hang Yuan, Yuliang Wei, Bailing Wang, Weizhen Qi, and Kai Chen. 2025. Long-short chain-of-thought mixture supervised fine-tuning eliciting efficient reasoning in large language models. *Preprint*, arXiv:2505.03469.

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. 2025. A survey on test-time scaling in large language models: What, how, where, and how well? *Preprint*, arXiv:2503.24235.

Han Zhao, Haotian Wang, Yiping Peng, Sitong Zhao, Xiaoyu Tian, Shuaiting Chen, Yunjie Ji, and Xiangang Li. 2025. 1.4 million open-source distilled reasoning dataset to empower large language model training. *Preprint*, arXiv:2503.19633.

A Prompt Template

A.1 Prompt Template For Training

Supervised Fine Tune

Given the following problem, solve it step by step.

QUESTION: {question}

<think> {thought_process} </think>

{Final Answer}

A.2 Prompt Template for Obtaining High-Quality Corpus

Compress the given reasoning steps to short expressions, and such that you (Deepseek) can understand reasoning and reconstruct it as close as possible to the original.
Unlike the usual text compression, I need you to comply with the 5 conditions below:

1. You can ONLY remove unimportant words.
2. Do not reorder the original words.
3. Do not change the original words.
4. Do not use abbreviations or emojis.
5. Do not add new words or symbols.

Compress the origin aggressively by removing words only. Compress the origin as short as you can, while retaining as much information as possible. If you understand, please compress the following reasoning steps:

{reasoning_steps}

The compressed reasoning steps are:

B Additional Experimental Results

Below we present the results of different compression methods on Llama3.1-8B-Instruct model.

Table 6: Experimental results of various compression methods on **Llama3.1-8B**, showing accuracy, average reasoning CoT tokens, and compression ratio (actual ratio).

Methods	Ratio (Actual)	MATH500		AIME24		GPQA Diamond	
		Accuracy ↑	Tokens ↓	Accuracy ↑	Tokens ↓	Accuracy ↑	Tokens ↓
Original	1.0	65.0					
TokenSkip	0.9(0.88)	56.4	11985	0	27882	35.8	17846
	0.8(0.80)	51.2	13145	3.3	30443	30.3	17960
	0.7(0.71)	44.6	14354	3.3	34249	27.7	19265
	0.6(0.62)	32.4	15453	0	23319	29.7	20800
	0.5(0.50)	23.3	16013	0	22318	26.7	22010
CTS	0.9(0.87)	60.6	12047	3.3	29144	32.3	18503
	0.8(0.81)	58.4	12134	6.7	24906	40.4	18981
	0.7(0.74)	55.0	9987	3.3	25933	32.3	16571
	0.6(0.66)	50.8	2808	0	3781	26.7	3492
	0.5(0.58)	45.5	2625	0	3478	29.2	3080